
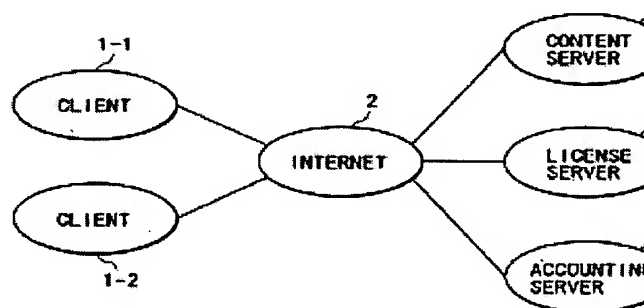


INFORMATION PROCESSOR**Publication number:** WO02080067**Publication date:** 2002-10-10**Inventor:** TANAKA KOICHI (JP); ISHIGURO RYUJI (JP)**Applicant:** SONY CORP (JP); TANAKA KOICHI (JP); ISHIGURO RYUJI (JP)**Classification:****- International:** G06F21/00; G06Q10/00; G06Q30/00; G06F21/00;
G06Q10/00; G06Q30/00; (IPC1-7): G06F17/60**- European:** G06F21/00N7D; G06Q10/00F; G06Q30/00C**Application number:** WO2002JP02959 20020327**Priority number(s):** JP20010094810 20010329**Also published as:** US2003182236 (A)**Report a data error he**

Abstract not available for WO02080067

Abstract of corresponding document: **US2003182236**

This invention relates to an information processing apparatus for permitting relatively unrestrained distribution of contents while managing the contents individually. Contents are separated from licenses granting the right to their use. To each content are added use conditions specifying how the content is to be used. The content is furnished with a mark which comprises a leaf ID, an ownership flag, a use start timestamp, and a copy count. This invention may be applied to any apparatus for offering contents.

Data supplied from the **esp@cenet** database - Worldwide

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2002 年 10 月 10 日 (10.10.2002)

PCT

(10) 国際公開番号
WO 02/080067 A2

- (51) 国際特許分類: G06F 17/60 北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 石黒 隆二 (ISHIGURO, Ryuji) [JP/JP]; 〒141-0001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP).
- (21) 国際出願番号: PCT/JP02/02959
- (22) 国際出願日: 2002 年 3 月 27 日 (27.03.2002)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ: 特願 2001-94810 2001 年 3 月 29 日 (29.03.2001) JP
- (74) 代理人: 稲本 義雄 (INAMOTO, Yoshio); 〒160-0023 東京都新宿区西新宿 7 丁目 1 1 番 1 8 号 7 1 1 ビルディング 4 階 Tokyo (JP).
- (81) 指定国 (国内): JP, KR, US.
- (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP).
- 添付公開書類:
— 第 17 条(2)(a)に基づく宣言; 要約なし; 国際調査機関により点検されていない発明の名称。
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 田中 浩一 (TANAKA, Koichi) [JP/JP]; 〒141-0001 東京都品川区
- 2 文字コード及び他の略語については、定期発行される各 PCT ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: INFORMATION PROCESSOR

(54) 発明の名称: 情報処理装置

(57) Abstract:



WO 02/080067 A2

明細書

情報処理装置

技術分野

- 5 本発明は、情報処理装置に関し、特に、コンテンツと、それを使用するライセンスとを分離してコンテンツを管理する場合において、コンテンツを個々に管理できるようにする情報処理装置に関する。

背景技術

- 10 最近、インターネットが普及し、オーディオやビデオなどの各種のコンテンツが、インターネットを介して伝送されるようになってきた。

このように、コンテンツがインターネットを介して伝送されるようになると、その規模が世界的であるため、コンテンツの著作権を、確実に管理できるようにすることが要求される。

- 15 しかしながら、従来の著作権を管理する方法は、コンテンツの不正なコピーを防止することに重点がおかれるあまり、コンテンツそのものの配布が困難になってしまう課題があった。

発明の開示

- 20 本発明は、このような状況に鑑みてなされたものであり、コンテンツを比較的自由に流通させつつ、その著作権を確実に管理できるようにするものである。

本発明の情報処理装置は、クライアントからライセンスを指定する指定情報を取得する第1の取得手段と、第1の取得手段により取得された指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得手段と、第2の
25 取得手段により取得された使用状況情報に対応するマーク情報を生成し、クライアントに提供する提供手段とを備えることを特徴とする。

前記マーク情報は、ライセンスを識別する識別情報、ライセンスの買い取りを表す情報、ライセンスが対象とするコンテンツの使用を開始した日時、またはコンテンツをコピーした回数を含むようにすることができる。

本発明の情報処理方法は、クライアントからライセンスを指定する指定情報を取得する第1の取得ステップと、第1の取得ステップの処理により取得された指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得ステップと、第2の取得ステップの処理により取得された使用状況情報に対応するマーク情報を生成し、クライアントに提供する提供ステップとを含むことを特徴とする。

- 10 本発明の記録媒体のプログラムは、クライアントからライセンスを指定する指定情報を取得する第1の取得ステップと、第1の取得ステップの処理により取得された指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得ステップと、第2の取得ステップの処理により取得された使用状況情報に対応するマーク情報を生成し、クライアントに提供する提供ステップとを含むことを特徴とする。

- 15 本発明のプログラムは、クライアントからライセンスを指定する指定情報を取得する第1の取得ステップと、第1の取得ステップの処理により取得された指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得ステップと、第2の取得ステップの処理により取得された使用状況情報に対応するマーク情報を生成し、クライアントに提供する提供ステップとをコンピュータに実現させる。

本発明においては、ライセンスに関する使用状況情報に対応するマーク情報が生成され、クライアントに提供される。

25 図面の簡単な説明

図1は、本発明を適用したコンテンツ提供システムの構成を示すブロック図である。

図 2 は、図 1 のクライアントの構成を示すブロック図である。

図 3 は、図 1 のクライアントのコンテンツのダウンロード処理を説明するフローチャートである。

図 4 は、図 1 のコンテンツサーバのコンテンツ提供処理を説明するフローチャートである。

図 5 は、図 4 のステップ S 2 6 におけるフォーマットの例を示す図である。

図 6 は、図 1 のクライアントのコンテンツ再生処理を説明するフローチャートである。

図 7 は、図 6 のステップ S 4 3 のライセンス取得処理の詳細を説明するフローチャートである。

図 8 は、ライセンスの構成を示す図である。

図 9 は、図 1 のライセンスサーバのライセンス提供の処理を説明するフローチャートである。

図 1 0 は、図 6 のステップ S 4 5 におけるライセンス更新処理の詳細を説明するフローチャートである。

図 1 1 は、図 1 のライセンスサーバのライセンス更新処理を説明するフローチャートである。

図 1 2 は、キーの構成を説明する図である。

図 1 3 は、カテゴリノードを説明する図である。

図 1 4 は、ノードとデバイスの対応の具体例を示す図である。

図 1 5 A は、有効化キーブロックの構成を説明する図である。

図 1 5 B は、有効化キーブロックの構成を説明する図である。

図 1 6 は、有効化キーブロックの利用を説明する図である。

図 1 7 は、有効化キーブロックのフォーマットの例を示す図である。

図 1 8 は、有効化キーブロックのタグの構成を説明する図である。

図 1 9 は、DNK を用いたコンテンツの復号処理を説明する図である。

図 2 0 は、有効化キーブロックの例を示す図である。

図 2 1 は、複数のコンテンツの 1 つのデバイスに対する割り当てを説明する図である。

図 2 2 は、ライセンスのカテゴリを説明する図である。

図 2 3 は、登録処理を説明するタイミングチャートである。

5 図 2 4 は、クライアントのリッピング処理を説明するフローチャートである。

図 2 5 は、ウォーターマークの構成を説明する図である。

図 2 6 は、コンテンツのフォーマットの例を示す図である。

図 2 7 は、公開鍵証明書¹の例を示す図である。

図 2 8 は、コンテンツの配布を説明する図である。

10 図 2 9 は、クライアントのコンテンツのチェックアウト処理を説明するフローチャートである。

図 3 0 は、タグによる有効化キーブロックをたどる例を説明する図である。

図 3 1 は、有効化キーブロックの構成例を示す図である。

図 3 2 は、マークの構成を説明する図である。

15 図 3 3 は、クライアントのライセンス買い取り処理を説明するフローチャートである。

図 3 4 は、ライセンスサーバのライセンス買い取り処理を説明するフローチャートである。

図 3 5 は、マークの構成例を示す図である。

20 図 3 6 は、クライアントの証明書の登録処理を説明するフローチャートである。

図 3 7 は、コンテンツサーバの証明書登録処理を説明するフローチャートである。

図 3 8 は、グループの証明書の例を示す図である。

25 図 3 9 は、グルーピングが行われている場合におけるコンテンツサーバの処理を説明するフローチャートである。

図 4 0 は、コンテンツキーの暗号化の例を示す図である。

図4 1は、グループに属するクライアントの処理を説明するフローチャートである。

図4 2は、他のクライアントにライセンスをチェックアウトするクライアントの処理を説明するフローチャートである。

- 5 図4 3は、他のクライアントからライセンスのチェックアウトを受けるクライアントの処理を説明するフローチャートである。

図4 4は、ライセンスのチェックアウトを受けたクライアントの再生処理を説明するフローチャートである。

- 10 図4 5は、他のクライアントからライセンスのチェックインを受けるクライアントの処理を説明するフローチャートである。

図4 6は、他のクライアントにライセンスをチェックインするクライアントの処理を説明するフローチャートである。

図4 7は、MACの生成を説明する図である。

図4 8は、ICV生成キーの復号処理を説明するフローチャートである。

- 15 図4 9は、ICV生成キーの他の復号処理を説明する図である。

図5 0 Aは、ICVによるライセンスのコピーの管理を説明する図である。

図5 0 Bは、ICVによるライセンスのコピーの管理を説明する図である。

図5 1は、ライセンスの管理を説明する図である。

20 発明を実施するための最良の形態

- 図1は、本発明を適用したコンテンツ提供システムの構成を示している。インターネット2には、クライアント1-1, 1-2（以下、これらのクライアントを個々に区別する必要がない場合、単にクライアント1と称する）が接続されている。この例においては、クライアントが2台のみ示されているが、インターネット2には、任意の台数のクライアントが接続される。
- 25

また、インターネット2には、クライアント1に対してコンテンツを提供するコンテンツサーバ3、コンテンツサーバ3が提供するコンテンツを利用するのに

必要なライセンスをクライアント 1 に対して付与するライセンスサーバ 4、およびクライアント 1 がライセンスを受け取った場合に、そのクライアント 1 に対して課金処理を行う課金サーバ 5 が接続されている。

これらのコンテンツサーバ 3、ライセンスサーバ 4、および課金サーバ 5 も、

- 5 任意の台数、インターネット 2 に接続される。

図 2 はクライアント 1 の構成を表している。

- 図 2 において、CPU (Central Processing Unit) 2 1 は、ROM (Read Only Memory) 2 2 に記憶されているプログラム、または記憶部 2 8 から RAM (Random Access Memory) 2 3 にロードされたプログラムに従って各種の処理を実行する。タイマ 2 0 は、計時動作を行い、時刻情報を CPU 2 1 に供給する。RAM 2 3 にはまた、CPU 2 1 が各種の処理を実行する上において必要なデータなども適宜記憶される。
- 10

- 暗号化復号部 2 4 は、コンテンツデータを暗号化するとともに、既に暗号化されているコンテンツデータを復号する処理を行う。コーデック部 2 5 は、例えば、
- 15 ATRAC (Adaptive Transform Acoustic Coding) 3 方式などでコンテンツデータをエンコードし、入出力インタフェース 3 2 を介してドライブ 3 0 に接続されている半導体メモリ 4 4 に供給し、記録させる。あるいはまた、コーデック部 2 5 は、ドライブ 3 0 を介して半導体メモリ 4 4 より読み出した、エンコードされているデータをデコードする。

- 20 半導体メモリ 4 4 は、例えば、メモリスティック (商標) などにより構成される。

CPU 2 1、ROM 2 2、RAM 2 3、暗号化復号部 2 4、およびコーデック部 2 5 は、バス 3 1 を介して相互に接続されている。このバス 3 1 にはまた、入出力インタフェース 3 2 も接続されている。

- 25 入出力インタフェース 3 2 には、キーボード、マウスなどよりなる入力部 2 6、CRT、LCD などよりなるディスプレイ、並びにスピーカなどよりなる出力部 2 7、ハードディスクなどより構成される記憶部 2 8、モデム、ターミナルアダプ

タなどより構成される通信部 29 が接続されている。通信部 29 は、インターネット 2 を介しての通信処理を行う。通信部 29 はまた、他のクライアントとの間で、アナログ信号またはデジタル信号の通信処理を行う。

入出力インタフェース 32 にはまた、必要に応じてドライブ 30 が接続され、
5 磁気ディスク 41、光ディスク 42、光磁気ディスク 43、或いは半導体メモリ 44 などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 28 にインストールされる。

なお、図示は省略するが、コンテンツサーバ 3、ライセンスサーバ 4、課金サーバ 5 も、図 2 に示したクライアント 1 と基本的に同様の構成を有するコンピュータにより構成される。そこで、以下の説明においては、図 2 の構成は、コンテンツサーバ 3、ライセンスサーバ 4、課金サーバ 5 などの構成としても引用される。
10

次に、図 3 のフローチャートを参照して、クライアント 1 がコンテンツサーバ 3 からコンテンツの提供を受ける処理について説明する。

15 ユーザが、入力部 26 を操作することでコンテンツサーバ 3 に対するアクセスを指令すると、CPU 21 は、ステップ S1 において、通信部 29 を制御し、インターネット 2 を介してコンテンツサーバ 3 にアクセスさせる。ステップ S2 において、ユーザが、入力部 26 を操作して、提供を受けるコンテンツを指定すると、CPU 21 は、この指定情報を受け取り、通信部 29 から、インターネット
20 2 を介してコンテンツサーバ 3 に、指定されたコンテンツを通知する。図 4 のフローチャートを参照して後述するように、この通知を受けたコンテンツサーバ 3 は、暗号化されたコンテンツデータを送信してくるので、ステップ S3 において、CPU 21 は、通信部 29 を介して、このコンテンツデータを受信すると、ステップ S4 において、その暗号化されているコンテンツデータを記憶部 28 を構成
25 するハードディスクに供給し、記憶させる。

次に、図 4 のフローチャートを参照して、クライアント 1 の以上の処理に対応するコンテンツサーバ 3 のコンテンツ提供処理について説明する。なお、以下の

説明において、図 2 のクライアント 1 の構成は、コンテンツサーバ 3 の構成としても引用される。

ステップ S 2 1 において、コンテンツサーバ 3 の CPU 2 1 は、インターネット 2 から通信部 2 9 を介してクライアント 1 よりアクセスを受けるまで待機し、
5 アクセスを受けたと判定したとき、ステップ S 2 2 に進み、クライアント 1 から送信されてきたコンテンツを指定する情報を取り込む。このコンテンツを指定する情報は、クライアント 1 が、図 3 のステップ S 2 において通知してきた情報である。

ステップ S 2 3 において、コンテンツサーバ 3 の CPU 2 1 は、記憶部 2 8 に
10 記憶されているコンテンツデータの中から、ステップ S 2 2 の処理で取り込まれた情報で指定されたコンテンツを読み出す。CPU 2 1 は、ステップ S 2 4 において、記憶部 2 8 から読み出されたコンテンツデータを、暗号化復号部 2 4 に供給し、コンテンツキー Kc を用いて暗号化させる。

記憶部 2 8 に記憶されているコンテンツデータは、コーデック部 2 5 により、
15 既に ATRAC 3 方式によりエンコードされているので、このエンコードされているコンテンツデータが暗号化されることになる。

なお、もちろん、記憶部 2 8 に予め暗号化した状態でコンテンツデータを記憶させることができる。この場合には、ステップ S 2 4 の処理は省略することが可能である。

20 次に、ステップ S 2 5 において、コンテンツサーバ 3 の CPU 2 1 は、暗号化したコンテンツデータを伝送するフォーマットを構成するヘッダに、暗号化されているコンテンツを復号するのに必要なキー情報（図 5 を参照して後述する EKB (Enabling Key Block) と K_{EKBC} (Kc)) と、コンテンツを利用するのに必要なライセンスを識別するためのライセンス ID を付加する。そして、ステ
25 ップ S 2 6 において、コンテンツサーバ 3 の CPU 2 1 は、ステップ S 2 4 の処理で暗号化したコンテンツと、ステップ S 2 5 の処理でキーとライセンス ID を

付加したヘッダとをフォーマット化したデータを、通信部 29 から、インターネット 2 を介して、アクセスしてきたクライアント 1 に送信する。

- 図 5 は、このようにして、コンテンツサーバ 3 からクライアント 1 にコンテンツが供給される場合のフォーマットの構成を表している。同図に示されるように、
- 5 このフォーマットは、ヘッダ (Header) とデータ (Data) とにより構成される。

ヘッダには、コンテンツ情報 (Content information)、URL (Uniform Resource Locator)、ライセンス ID (License ID)、イネープリングキープブロック (有効化キープブロック) (EKB (Enabling Key Block)) および、EKB から生成されたキー K_{EKB} を用いて暗号化されたコンテンツキー K_c としてのデータ $K_{EKB}(K_c)$ が配置されている。なお、EKB については、図 15 A および図 15 B を参照して後述する。

10

コンテンツ情報には、データとしてフォーマット化されているコンテンツデータを識別するための識別情報としてのコンテンツ ID (CID)、そのコンテンツのコーデックの方式などの情報が含まれている。

- 15 URL は、ライセンス ID で規定されるライセンスを取得するときアクセスするアドレス情報であり、図 1 のシステムの場合、具体的には、ライセンスを受けるために必要なライセンスサーバ 4 のアドレスである。ライセンス ID は、データとして記録されているコンテンツを利用するとき必要とされるライセンスを識別するものである。

- 20 データは、任意の数の暗号化ブロック (Encryption Block) により構成される。各暗号化ブロックは、イニシャルベクトル (IV (Initial Vector))、シード (Seed)、およびコンテンツデータをキー K'_c で暗号化したデータ $E_{K'_c}(\text{data})$ により構成されている。

- キー K'_c は、次式により示されるように、コンテンツキー K_c と、乱数で設定される値 Seed をハッシュ関数に適用して演算された値により構成される。
- 25

$$K'_c = \text{Hash}(K_c, \text{Seed})$$

イニシャルベクトル IV とシード Seed は、各暗号化ブロック毎に異なる値に設定される。

この暗号化は、コンテンツのデータを 8 バイト単位で区分して、8 バイト毎に行われる。後段の 8 バイトの暗号化は、前段の 8 バイトの暗号化の結果を利用して行われる CBC (Cipher Block Chaining) モードで行われる。

CBC モードの場合、最初の 8 バイトのコンテンツデータを暗号化するとき、その前段の 8 バイトの暗号化結果が存在しないため、最初の 8 バイトのコンテンツデータを暗号化するときは、イニシャルベクトル IV を初期値として暗号化が行われる。

10 この CBC モードによる暗号化を行うことで、1 つの暗号化ブロックが解読されたとしても、その影響が、他の暗号化ブロックにおよぶことが抑制される。

なお、この暗号化については、図 4 7 を参照して、後に詳述する。

また、暗号方式についてはこれに限らず、単にコンテンツキー Kc でコンテンツデータを暗号化しても良い。

15 以上のようにして、クライアント 1 は、コンテンツサーバ 3 からコンテンツを無料で、自由に取得することができる。従って、コンテンツそのものは、大量に、配布することが可能となる。

しかしながら、各クライアント 1 は、取得したコンテンツを利用するとき、ライセンスを保持している必要がある。そこで、図 6 を参照して、クライアント 1

20 がコンテンツを再生する場合の処理について説明する。

ステップ S 4 1 において、クライアント 1 の CPU 2 1 は、ユーザが入力部 2 6 を操作することで指示したコンテンツの識別情報 (CID) を取得する。この識別情報は、例えば、コンテンツのタイトルや、記憶されている各コンテンツ毎に付与されている番号などにより構成される。

25 そして、CPU 2 1 は、コンテンツが指示されると、そのコンテンツに対応するライセンス ID (そのコンテンツを使用するのに必要なライセンスの ID) を読

み取る。このライセンス ID は、図 5 に示されるように、暗号化されているコンテンツデータのヘッダに記述されているものである。

次に、ステップ S 4 2 に進み、CPU 2 1 は、ステップ S 4 1 で読み取られたライセンス ID に対応するライセンスが、クライアント 1 により既に取得され、
5 記憶部 2 8 に記憶されているか否かを判定する。まだ、ライセンスが取得されていない場合には、ステップ S 4 3 に進み、CPU 2 1 は、ライセンス取得処理を実行する。このライセンス取得処理の詳細は、図 7 のフローチャートを参照して後述する。

ステップ S 4 2 において、ライセンスが既に取得されていると判定された場合、
10 または、ステップ S 4 3 において、ライセンス取得処理が実行された結果、ライセンスが取得された場合、ステップ S 4 4 に進み、CPU 2 1 は、取得されているライセンスは有効期限内のものであるか否かを判定する。ライセンスが有効期限内のものであるか否かは、ライセンスの内容として規定されている期限（後述する図 8 参照）と、タイマ 2 0 により計時されている現在日時と比較することで
15 判断される。ライセンスの有効期限が既に満了していると判定された場合、CPU 2 1 は、ステップ S 4 5 に進み、ライセンス更新処理を実行する。このライセンス更新処理の詳細は、図 1 0 のフローチャートを参照して後述する。

ステップ S 4 4 において、ライセンスはまだ有効期限内であると判定された場合、または、ステップ S 4 5 において、ライセンスが更新された場合、ステップ
20 S 4 6 に進み、CPU 2 1 は、暗号化されているコンテンツデータを記憶部 2 8 から読み出し、RAM 2 3 に格納させる。そして、ステップ S 4 7 において、CPU 2 1 は、RAM 2 3 に記憶された暗号化ブロックのデータを、図 5 のデータに配置されている暗号化ブロック単位で、暗号化復号部 2 4 に供給し、コンテンツキー Kc を用いて復号させる。

25 コンテンツキー Kc を得る方法の具体例は、図 1 5 A および図 1 5 B を参照して後述するが、デバイスノードキー（DNK(Device Node Key)）を用いて、

EKB（図5）に含まれるキー K_{EKBC} を得ることができ、そのキー K_{EKBC} を用いて、データ K_{EKBC} （ K_c ）（図5）から、コンテンツキー K_c を得ることができる。

- CPU 21は、さらに、ステップS 48において、暗号化復号部24により復号されたコンテンツデータをコーデック部25に供給し、デコードさせる。そして、コーデック部25によりデコードされたデータを、CPU 21は、入出力インタフェース32から出力部27に供給し、D/A変換させ、スピーカから出力させる。

次に、図7のフローチャートを参照して、図6のステップS 43で行われるライセンス取得処理の詳細について説明する。

- 10 クライアント1は、事前にライセンスサーバにアクセスして登録処理を行うことにより、リーフID、DNK(Device Node Key)、クライアント1の秘密鍵・公開鍵のペア、ライセンスサーバの公開鍵、及び各公開鍵の証明書を含むサービスデータを取得しておく。クライアントの登録処理の詳細は図23を参照して後述する。
- 15 リーフIDは、クライアント毎に割り当てられた識別情報を表し、DNKは、そのライセンスに対応するEKB（有効化キーブロック）に含まれる暗号化されているコンテンツキー K_c を復号するのに必要なデバイスノードキーである（図12を参照して後述する）。

- 最初にステップS 61において、CPU 21は、いま処理対象とされているライセンスIDに対応するURLを、図5に示すヘッダから取得する。上述したように、このURLは、やはりヘッダに記述されているライセンスIDに対応するライセンスを取得するときアクセスすべきアドレスである。そこで、ステップS 62において、CPU 21は、ステップS 61で取得したURLにアクセスする。具体的には、通信部29によりインターネット2を介してライセンスサーバ4に
- 20 アクセスが行われる。このとき、ライセンスサーバ4は、クライアント1に対して、購入するライセンス（コンテンツを使用するのに必要なライセンス）を指定するライセンス指定情報、並びにユーザIDとパスワードの入力を要求してくる
- 25

(後述する図 9 のステップ S 1 0 2)。CPU 2 1 は、この要求を出力部 2 7 の表示部に表示させる。ユーザは、この表示に基づいて、入力部 2 6 を操作して、ライセンス指定情報、ユーザ ID、およびパスワードを入力する。なお、このユーザ ID とパスワードは、クライアント 1 のユーザが、インターネット 2 を介してライセンスサーバ 4 にアクセスし、事前を取得しておいたものである。

CPU 2 1 は、ステップ S 6 3, S 6 4 において、入力部 2 6 から入力されたライセンス指定情報を取り込むとともに、ユーザ ID とパスワードを取り込む。CPU 2 1 は、ステップ S 6 5 において、通信部 2 9 を制御し、入力されたユーザ ID とパスワード、ライセンス指定情報、並びにサービスデータ (後述する) に含まれるリーフ ID を含むライセンス要求を、インターネット 2 を介してライセンスサーバ 4 に送信させる。

ライセンスサーバ 4 は、図 9 を参照して後述するように、ユーザ ID とパスワード、並びにライセンス指定情報に基づいてライセンスを送信してくる (ステップ S 1 0 9) か、または、条件が満たされない場合には、ライセンスを送信してこない (ステップ S 1 1 2)。

ステップ S 6 6 において、CPU 2 1 は、ライセンスサーバ 4 からライセンスが送信されてきたか否かを判定し、ライセンスが送信されてきた場合には、ステップ S 6 7 に進み、そのライセンスを記憶部 2 8 に供給し、記憶させる。

ステップ S 6 6 において、ライセンスが送信されて来ないと判定した場合、CPU 2 1 は、ステップ S 6 8 に進み、エラー処理を実行する。具体的には、CPU 2 1 は、コンテンツを利用するためのライセンスが得られないので、コンテンツの再生処理を禁止する。

以上のようにして、各クライアント 1 は、コンテンツデータに付随しているライセンス ID に対応するライセンスを取得して、初めて、そのコンテンツを使用することが可能となる。

なお、図 7 のライセンス取得処理は、各ユーザがコンテンツを取得する前に、予め行っておくようにすることも可能である。

クライアント 1 に提供されるライセンスは、例えば、図 8 に示されるように、使用条件、リーフ ID 等を含んでいる。

使用条件には、そのライセンスに基づいて、コンテンツを使用することが可能な使用期限、そのライセンスに基づいて、コンテンツをダウンロードすることが可能なダウンロード期限、そのライセンスに基づいて、コンテンツをコピーすることが可能な回数（許されるコピー回数）、チェックアウト回数、最大チェックアウト回数、そのライセンスに基づいて、コンテンツを CD-R に記録することができる権利、PD（Portable Device）にコピーすることが可能な回数、ライセンスを所有権（買い取り状態）に移行できる権利、使用ログをとる義務等を示す情報が含まれる。

次に、図 9 のフローチャートを参照して、図 7 のクライアント 1 のライセンス取得処理に対応して実行されるライセンスサーバ 4 のライセンス提供処理について説明する。なお、この場合においても、図 2 のクライアント 1 の構成は、ライセンスサーバ 4 の構成として引用される。

ステップ S 1 0 1 において、ライセンスサーバ 4 の CPU 2 1 は、クライアント 1 よりアクセスを受けるまで待機し、アクセスを受けたとき、ステップ S 1 0 2 に進み、アクセスしてきたクライアント 1 に対して、ユーザ ID とパスワード、並びに、ライセンス指定情報の送信を要求する。上述したようにして、クライアント 1 から、図 7 のステップ S 6 5 の処理で、ユーザ ID とパスワード、リーフ ID 並びにライセンス指定情報（ライセンス ID）が送信されてきたとき、ライセンスサーバ 4 の CPU 2 1 は、通信部 2 9 を介してこれを受信し、取り込む処理を実行する。

そして、ライセンスサーバ 4 の CPU 2 1 は、ステップ S 1 0 3 において、通信部 2 9 から課金サーバ 5 にアクセスし、ユーザ ID とパスワードに対応するユーザの与信処理を要求する。課金サーバ 5 は、インターネット 2 を介してライセンスサーバ 4 から与信処理の要求を受けると、そのユーザ ID とパスワードに対応するユーザの過去の支払い履歴などを調査し、そのユーザが、過去にライセン

スの対価の不払いの実績があるか否かなどを調べ、そのような実績がない場合には、ライセンスの付与を許容する与信結果を送信し、不払いの実績などがある場合には、ライセンス付与の不許可の与信結果を送信する。

5 ステップS 1 0 4において、ライセンスサーバ4のCPU 2 1は、課金サーバ5からの与信結果が、ライセンスを付与することを許容する与信結果であるか否かを判定し、ライセンスの付与が許容されている場合には、ステップS 1 0 5に進み、ステップS 1 0 2の処理で取り込まれたライセンス指定情報に対応するライセンスを、記憶部2 8に記憶されているライセンスの中から取り出す。記憶部2 8に記憶されているライセンスは、あらかじめライセンスID、バージョン、
10 作成日時、有効期限等の情報が記述されている。ステップS 1 0 6において、CPU 2 1は、そのライセンスに受信したリーフIDを付加する。さらに、ステップS 1 0 7において、CPU 2 1は、ステップS 1 0 5で選択されたライセンスに対応づけられている使用条件を選択する。あるいはまた、ステップS 1 0 2の処理で、ユーザから使用条件が指定された場合には、その使用条件が必要に応じて、予め用意されている使用条件に付加される。CPU 2 1は、選択された使用
15 条件をライセンスに付加する。

ステップS 1 0 8において、CPU 2 1はライセンスサーバの秘密鍵によりライセンスに署名し、これにより、図8に示されるような構成のライセンスが生成される。

20 次に、ステップS 1 0 9に進み、ライセンスサーバ4のCPU 2 1は、そのライセンス（図8に示される構成を有する）を、通信部2 9からインターネット2を介してクライアント1に送信させる。

ステップS 1 1 0においてライセンスサーバ4のCPU 2 1は、ステップS 1 0 9の処理で、いま送信したライセンス（使用条件、リーフIDを含む）を、
25 ステップS 1 0 2の処理で取り込まれたユーザIDとパスワードに対応して、記憶部2 8に記憶させる。さらに、ステップS 1 1 1において、CPU 2 1は、課金処理を実行する。具体的には、CPU 2 1は、通信部2 9から課金サーバ5に、

そのユーザ ID とパスワードに対応するユーザに対する課金処理を要求する。課金サーバ 5 は、この課金の要求に基づいて、そのユーザに対する課金処理を実行する。上述したように、この課金処理に対して、そのユーザが支払いを行わなかったような場合には、以後、そのユーザは、ライセンスの付与を要求したとしても、ライセンスを受けることができないことになる。

すなわち、この場合には、課金サーバ 5 からライセンスの付与を不許可とする
与信結果が送信されてくるので、ステップ S 1 0 4 からステップ S 1 1 2 に進み、CPU 2 1 は、エラー処理を実行する。具体的には、ライセンスサーバ 4 の CPU 2 1 は、通信部 2 9 を制御してアクセスしてきたクライアント 1 に対して、ライ
10 センスを付与することができない旨のメッセージを出力し、処理を終了させる。

この場合、上述したように、そのクライアント 1 はライセンスを受けることができないので、そのコンテンツを利用すること（暗号を復号すること）ができないことになる。

図 1 0 は、図 6 のステップ S 4 5 におけるライセンス更新処理の詳細を表して
15 いる。図 1 0 のステップ S 1 3 1 乃至ステップ S 1 3 5 の処理は、図 7 のステップ S 6 1 乃至ステップ S 6 5 の処理と基本的に同様の処理である。ただし、ステップ S 1 3 3 において、CPU 2 1 は、購入するライセンスではなく、更新するライセンスのライセンス ID を取り込む。そして、ステップ S 1 3 5 において、CPU 2 1 は、ユーザ ID とパスワードとともに、更新するライセンスのライセン
20 ス ID を、ライセンスサーバ 4 に送信する。

ステップ S 1 3 5 の送信処理に対応して、ライセンスサーバ 4 は、後述するよう
に、使用条件を提示してくる（図 1 1 のステップ S 1 5 3）。そこで、クライアント 1 の CPU 2 1 は、ステップ S 1 3 6 において、ライセンスサーバ 4 から
の使用条件の提示を受信し、これを出力部 2 7 に出力し、表示させる。ユーザは、
25 入力部 2 6 を操作して、この使用条件の中から所定の使用条件を選択したり、所
定の使用条件を新たに追加したりする。ステップ S 1 3 7 で CPU 2 1 は、以上
のようにして選択された使用条件（ライセンスを更新する条件）を購入するため

の申し込みをライセンスサーバ4に送信する。この申し込みに対応して、後述するようにライセンスサーバ4は、最終的な使用条件を送信してくる（図11のステップS154）。そこで、ステップS138において、クライアント1のCPU21は、ライセンスサーバ4からの使用条件を取得し、ステップS139において、その使用条件を記憶部28にすでに記憶されている対応するライセンスの使用条件として更新する。

図11は、以上のクライアント1のライセンス更新処理に対応して、ライセンスサーバ4が実行するライセンス更新処理を表している。

最初に、ステップS151において、ライセンスサーバ4のCPU21は、クライアント1からのアクセスを受けると、ステップS152において、クライアント1がステップS135で送信したライセンス指定情報をライセンス更新要求情報とともに受信する。

ステップS153において、CPU21は、ライセンスの更新要求を受信すると、そのライセンスに対応する使用条件（更新する使用条件）を、記憶部28から読み出し、クライアント1に送信する。

この提示に対して、上述したように、クライアント1から使用条件の購入が図10のステップS137の処理で申し込まれると、ステップS154において、ライセンスサーバ4のCPU21は、申し込まれた使用条件に対応するデータを生成し、ステップS154において、クライアントと1に送信する。クライアント1は、上述したように、ステップS139の処理で受信した使用条件を用いて、すでに登録されているライセンスの使用条件を更新する。

本発明においては、図12に示されるように、ブロードキャストインクリプション（Broadcast Encryption）方式の原理に基づいて、デバイスとライセンスのキーが管理される（特開2001-352321号公報参照）。キーは、階層ツリー構造とされ、最下段のリーフ（leaf）が個々のデバイスのキーに対応する。図12の例の場合、番号0から番号15までの16個のデバイスまたはライセンスに対応するキーが生成される。

各キーは、図中丸印で示されるツリー構造の各ノードに対応して規定される。この例では、最上段のルートノードに対応してルートキーKR が、2 段目のノードに対応してキーK 0, K 1 が、3 段目のノードに対応してキーK 0 0 乃至K 1 1 が、第4 段目のノードに対応してキーK 0 0 0 乃至キーK 1 1 1 が、それぞれ
5 対応されている。そして、最下段のノードとしてのリーフ（デバイスノード）に、キーK 0 0 0 0 乃至K 1 1 1 1 が、それぞれ対応されている。

階層構造とされているため、例えば、キーK 0 0 1 0 とキー0 0 1 1 の上位のキーは、K 0 0 1 とされ、キーK 0 0 0 とキーK 0 0 1 の上位のキーは、K 0 0 とされている。以下同様に、キーK 0 0 とキーK 0 1 の上位のキーは、K 0 とされ、キーK 0 とキーK 1 の上位のキーは、KR とされている。
10

コンテンツを利用するキーは、最下段のリーフから、最上段のルートノードまでの1 つのパスの各ノードに対応するキーで管理される。例えば、番号3 のノード（リーフ ID）に対応するライセンスに基づき、コンテンツを利用するキーは、キーK 0 0 1 1, K 0 0 1, K 0 0, K 0, KR を含むパスの各キーで管理され
15 る。

本発明のシステムにおいては、図1 3 に示されるように、図1 2 の原理に基づいて構成されるキーシステムで、デバイスのキーとライセンスのキーの管理が行われる。図1 3 の例では、8 + 2 4 + 3 2 段のノードがツリー構造とされ、ルートノードから下位の8 段までの各ノードにカテゴリが対応される。ここにおける
20 カテゴリとは、例えばメモリスティックなどの半導体メモリを使用する機器のカテゴリ、デジタル放送を受信する機器のカテゴリといったカテゴリを意味する。そして、このカテゴリノードのうちの1 つのノードに、ライセンスを管理するシステムとして本システム（T システムと称する）が対応する。

すなわち、このT システムのノードよりさらに下の階層の2 4 段のノードに対応するキーにより、ライセンスが対応される。この例の場合、これにより、2 の
25 24 乗（約1 6 メガ）のライセンスを規定することができる。さらに、最も下側の3 2 段の階層により、2 の32 乗（約4 ギガ）のユーザ（あるいはクライアント

ト 1) を規定することができる。最下段の 3 2 段のノードに対応するリーフからルートノードまでのパスの各ノードに対応するキーが、DNK (Device Node Key) を構成し、最下段のリーフに対応する ID がリーフ ID とされる。

各デバイスやライセンスのキーは、6 4 (= 8 + 2 4 + 3 2) 段の各ノードで
5 構成されるパスの内の 1 つに対応される。例えば、コンテンツを暗号化したコンテンツキーは、対応するライセンスに割り当てられたパスを構成するノードに対応するキーを用いて暗号化される。上位の階層のキーは、その直近の下位の階層のキーを用いて暗号化され、E K B (図 1 5 A および図 1 5 B を参照して後述する) 内に配置される。DNK は、E K B 内には配置されず、サービスデータに記述され、ユーザのクライアント 1 に与えられる。クライアント 1 は、サービスデータに記述されている DNK を用いて、コンテンツデータとともに配布される E K B (図 1 5 A および図 1 5 B) 内に記述されている直近の上位の階層のキーを復号し、復号して得たキーを用いて、E K B 内に記述されているさらにその上の階層のキーを復号する。以上の処理を順次行うことで、クライアント 1 は、その
10 パスに属するすべてのキーを得ることができる。

図 1 4 に階層ツリー構造のカテゴリの分類の具体的な例を示す。図 1 4 において、階層ツリー構造の最上段には、ルートキー KR 2 3 0 1 が設定され、以下の中間段にはノードキー 2 3 0 2 が設定され、最下段には、リーフキー 2 3 0 3 が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。
20

最上段から第 M 段目 (図 1 3 の例では、M = 8) の所定のノードがカテゴリノード 2 3 0 4 として設定される。すなわち第 M 段目のノードの各々が特定カテゴリのデバイス設定ノードとされる。第 M 段の 1 つのノードを頂点として M + 1 段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとされる。
25

例えば図 1 4 の第 M 段目の 1 つのノード 2 3 0 5 にはカテゴリ [メモリスティック (商標)] が設定され、このノード以下に連なるノード、リーフはメモリス

ティックを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード2305以下が、メモリスティックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義される。

さらに、M段から数段分下位の段をサブカテゴリノード2306として設定することができる。図14の例では、カテゴリ「メモリスティック」ノード2305の2段下のノードに、メモリスティックを使用したデバイスのカテゴリに含まれるサブカテゴリノードとして、「再生専用器」のノード2306が設定されている。さらに、サブカテゴリノードである再生専用器のノード2306以下に、再生専用器のカテゴリに含まれる音楽再生機能付き電話のノード2307が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる「PHS」ノード2308と、「携帯電話」ノード2309が設定されている。

さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えばあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば1つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器XYZ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器XYZに、その頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（EKB）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

このように、1つのノードを頂点として、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の1つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（EKB）を独自に生成して、頂点ノード以下に属するデバイスに配信す

る構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには全く影響を及ぼさずにキー更新を実行することができる。

例えば、図 1 2 に示されるツリー構造において、1 つのグループに含まれる 4 つのデバイス 0, 1, 2, 3 はノードキーとして共通のキー K_{00} 、 K_0 、 K_R を保有する。このノードキー共有構成を利用することにより、共通のコンテンツキーをデバイス 0, 1, 2, 3 のみに提供することが可能となる。たとえば、共通に保有するノードキー K_{00} 自体をコンテンツキーとして設定すれば、新たな鍵送付を実行することなくデバイス 0, 1, 2, 3 のみが共通のコンテンツキーの設定が可能である。また、新たなコンテンツキー K_{con} をノードキー K_{00} で暗号化した値 $Enc(K_{00}, K_{con})$ を、ネットワークを介してあるいは記録媒体に格納してデバイス 0, 1, 2, 3 に配布すれば、デバイス 0, 1, 2, 3 のみが、それぞれのデバイスにおいて保有する共有ノードキー K_{00} を用いて暗号 $Enc(K_{00}, K_{con})$ を解いてコンテンツキー K_{con} を得ることが可能となる。なお、 $Enc(K_a, K_b)$ は K_b を K_a によって暗号化したデータであることを示す。

また、ある時点 t において、デバイス 3 の所有する鍵 K_{0011} 、 K_{001} 、 K_{00} 、 K_0 、 K_R が攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス 0, 1, 2, 3 のグループ）で送受信されるデータを守るために、デバイス 3 をシステムから切り離す必要がある。そのためには、ノードキー K_{0011} 、 K_{00} 、 K_0 、 K_R を、それぞれ新たな鍵 $K(t)_{001}$ 、 $K(t)_{00}$ 、 $K(t)_0$ 、 $K(t)_R$ に更新し、デバイス 0, 1, 2 にその更新キーを伝える必要がある。ここで、 $K(t)_{aaa}$ は、鍵 K_{aaa} の世代 (Generation) t の更新キーであることを示す。

更新キーの配布処理について説明する。キーの更新は、例えば、図 1 5 A に示す有効化キーブロック (EKB: Enabling Key Block) と呼ばれるブロックデータによって構成されるテーブルを、ネットワークを介して、あるいは記録媒体に格納してデバイス 0, 1, 2 に供給することによって実行される。なお、有効化

キーブロック (EKB) は、図 1 2 に示されるようなツリー構造を構成する各リーフ (最下段のノード) に対応するデバイスに、新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック (EKB) は、キー更新ブロック (KRB : Key Renewal Block) と呼ばれることもある。

- 5 図 1 5 A に示す有効化キーブロック (EKB) は、ノードキーの更新に必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。図 1 5 A の例は、図 1 2 に示すツリー構造中のデバイス 0, 1, 2 において、世代 t の更新ノードキーを配布することを目的として形成されたブロックデータである。図 1 2 から明らかなように、デバイス 0, デバイス 1 は、更新ノードキーとして $K(t) 00$ 、 $K(t) 0$ 、 $K(t) R$ が必要であり、デバイス 2 は、更新ノードキーとして $K(t) 001$ 、 $K(t) 00$ 、 $K(t) 0$ 、 $K(t) R$ が必要である。
- 10

- 図 1 5 A の EKB に示されるように、EKB には複数の暗号化キーが含まれる。図 1 5 A の最下段の暗号化キーは、 $Enc(K0010, K(t) 001)$ である。これはデバイス 2 の持つリーフキー $K0010$ によって暗号化された更新ノードキー $K(t) 001$ であり、デバイス 2 は、自身の持つリーフキー $K0010$ によってこの暗号化キーを復号し、更新ノードキー $K(t) 001$ を得ることができる。また、復号により得た更新ノードキー $K(t) 001$ を用いて、図 1 5 A の下から 2 段目の暗号化キー $Enc(K(t) 001, K(t) 00)$ が復号可能となり、更新ノードキー $K(t) 00$ を得ることができる。
- 15
- 20

以下順次、図 1 5 A の上から 2 段目の暗号化キー $Enc(K(t) 00, K(t) 0)$ を復号することで、更新ノードキー $K(t) 0$ が得られ、これを用いて、図 1 5 A の上から 1 段目の暗号化キー $Enc(K(t) 0, K(t) R)$ を復号することで、更新ルートキー $K(t) R$ が得られる。

- 25 一方、ノードキー $K000$ は更新する対象に含まれておらず、ノード 0, 1 が、更新ノードキーとして必要なのは、 $K(t) 00$ 、 $K(t) 0$ 、 $K(t) R$ である。ノード 0, 1 は、デバイスノードキーに含まれるノードキー $K000$ を用い

て、図15Aの上から3段目の暗号化キー $Enc(K000, K(t)00)$ を復号することで更新ノードキー $K(t)00$ を取得し、以下順次、図15Aの上から2段目の暗号化キー $Enc(K(t)00, K(t)0)$ を復号することで、更新ノードキー $K(t)0$ を得、図15Aの上から1段目の暗号化キー Enc

- 5 $(K(t)0, K(t)R)$ を復号することで、更新ルートキー $K(t)R$ を得る。このようにして、デバイス0, 1, 2は更新したキー $K(t)R$ を得ることができる。

なお、図15Aのインデックスは、図の右側の暗号化キーを復号するための復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

- 10 図12に示すツリー構造の上位段のノードキー $K(t)0, K(t)R$ の更新が不要であり、ノードキー $K00$ のみの更新処理が必要である場合には、図15Bの有効化キーブロック(EKB)を用いることで、更新ノードキー $K(t)00$ をデバイス0, 1, 2に配布することができる。

- 図15Bに示すEKBは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図12に点線で示すグループ内のデバイス0, 1, 2, 3がある記録媒体を用いており、新たな共通のコンテンツキー $K(t)con$ が必要であるとする。このとき、デバイス0, 1, 2, 3の共通のノードキー $K00$ を更新した $K(t)00$ を用いて新たな共通の更新コンテンツキー $K(t)con$ を暗号化したデータ $Enc(K(t)00, K(t)con)$ が、図15Bに示されるEKBとともに配布される。この配布により、デバイス4など、その他のグループの機器が復号することができないデータとしての配布が可能となる。

- すなわち、デバイス0, 1, 2はEKBを処理して得たキー $K(t)00$ を用いて暗号文を復号すれば、 t 時点でのコンテンツキー $K(t)con$ を得ることが可能になる。

図16に、 t 時点でのコンテンツキー $K(t)con$ を得る処理例として、 $K(t)00$ を用いて新たな共通のコンテンツキー $K(t)con$ を暗号化したデ

ータ $E_{nc}(K(t)_{00}, K(t)_{con})$ と、図 15 B に示す EKB とを記録媒体を介して受領したデバイス 0 の処理を示す。すなわちこの例は、EKB による暗号化メッセージデータをコンテンツキー $K(t)_{con}$ とした例である。

図 16 に示すように、デバイス 0 は、記録媒体に格納されている世代 t 時点の EKB と、自分があらかじめ格納している DNK に含まれるノードキー K_{000} を用いて、上述した場合と同様の EKB 処理により、ノードキー $K(t)_{00}$ を生成する。さらに、デバイス 0 は、復号した更新ノードキー $K(t)_{00}$ を用いて、更新コンテンツキー $K(t)_{con}$ を復号して、後にそれを使用するために自分だけが持つリーフキー K_{0000} で暗号化して格納する。

図 17 に有効化キーブロック (EKB) のフォーマット例を示す。バージョン 601 は、有効化キーブロック (EKB) のバージョンを示す識別子である。なお、バージョンは、最新の EKB を識別する機能と、コンテンツとの対応関係を示す機能を持つ。デプスは、有効化キーブロック (EKB) の配布先のデバイスに対する階層ツリーの階層数を示す。データポイント 603 は、有効化キーブロック (EKB) 中のデータ部 606 の位置を示すポイントであり、タグポイント 604 はタグ部 607 の位置、署名ポイント 605 は署名 608 の位置を示すポイントである。

データ部 606 は、例えば更新するノードキーを暗号化したデータを格納する。例えば図 16 に示すような更新されたノードキーに関する各暗号化キー等を格納する。

タグ部 607 は、データ部 606 に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを、図 18 を用いて説明する。

図 18 では、データとして先に図 15 A で説明した有効化キーブロック (EKB) を送付する例を示している。この時のデータは、図 18 のテーブルに示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノードアドレスとする。この例の場合は、ルートキーの更新キー $K(t)_R$ が含ま

れているので、トップノードアドレスは KR となる。このとき、例えば最上段のデータ $Enc(K(t)0, K(t)R)$ は、図 18 に示す階層ツリーに示す位置 P0 に対応する。次の段のデータは、 $Enc(K(t)00, K(t)0)$ であり、ツリー上では前のデータの左下の位置 P00 に対応する。ツリー構造の

5 所定の位置から見て、その下に、データがある場合は、タグが 0、ない場合はタグが 1 に設定される。タグは {左 (L) タグ, 右 (R) タグ} として設定される。図 18 のテーブルの最上段のデータ $Enc(K(t)0, K(t)R)$ に対応する位置 PO の左下の位置 POO にはデータがあるので、L タグ = 0、右にはデータがないので、R タグ = 1 となる。以下、すべてのデータにタグが設定され、図

10 18 に示すデータ列、およびタグ列が構成される。

タグは、対応するデータ $Enc(Kxxx, Kyyy)$ が、ツリー構造のどこに位置しているのかを示すために設定されるものである。データ部 606 に格納されるキーデータ $Enc(Kxxx, Kyyy) \dots$ は、単純に暗号化されたキーの羅列データに過ぎないが、上述したタグによってデータとして格納された

15 暗号化キーのツリー上の位置が判別可能となる。上述したタグを用いずに、先の図 15 A および図 15 B で説明した構成のように、暗号化データに対応させたノード・インデックスを用いて、例えば、

0 : $Enc(K(t)0, K(t)R)$

00 : $Enc(K(t)00, K(t)0)$

20 000 : $Enc(K((t)000, K(t)00)$

・・・のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると、冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が

25 可能となる。

図 17 に戻って、EKB フォーマットについてさらに説明する。署名

(Signature) 608 は、有効化キーブロック (EKB) を発行した例えば鍵管

理センタ（ライセンスサーバ4）、コンテンツロバイダ（コンテンツサーバ3）、決済機関（課金サーバ5）等が実行する電子署名である。EKBを受領したデバイスは、署名検証によって正当な有効化キーブロック（EKB）発行者が発行した有効化キーブロック（EKB）であることを確認する。

- 5 以上のようにして、ライセンスサーバ4から供給されたライセンスに基づいて、コンテンツサーバ3から供給されたコンテンツを利用する処理をまとめると、図19に示されるようになる。

すなわち、コンテンツサーバ3からクライアント1に対してコンテンツが提供されるとともに、ライセンスサーバ4からクライアント1にライセンスが供給される。コンテンツは、コンテンツキーKcにより、暗号化されており（Enc（Kc, Content））、コンテンツキーKcは、ルートキーKR（EKBから得られるキーであって、図5におけるキーK_{EKBC}に対応する）で暗号化され（Enc（KR, Kc））、EKBとともに、暗号化されたコンテンツに付加されてクライアント1に提供される。

- 15 図19の例におけるEKBには、例えば、図20に示されるように、DNKで復号可能なルートキーKRが含まれている（Enc（DNK, KR））。従って、クライアント1は、サービスデータに含まれるDNKを利用して、EKBからルートキーKRを得ることができる。さらに、ルートキーKRを用いて、Enc（KR, Kc）からコンテンツキーKcを復号することができ、コンテンツキーKcを用いて、Enc（Kc, Content）からコンテンツを復号することができる。

このように、クライアント1にDNKを個別に割り当てることにより、図12、並びに図15Aおよび図15Bを参照して説明した原理に従って、個々のクライアント1のリボーク（revoke）が可能になる。

- 25 また、ライセンスにリーフIDを付加して配布することにより、クライアント1において、サービスデータとライセンスの対応付けが行われることになり、ライセンスの不正コピーを防止することが可能になる。

また、クライアント用の証明書と秘密鍵をサービスデータとして配信するようにすることで、エンドユーザも、これらを用いて不正コピーを防止可能なコンテンツを作成することが可能になる。

証明書と秘密鍵の利用については、図 2 9 のフローチャートを参照して後述する。

本発明においては、図 1 3 を参照して説明したように、カテゴリノードにライセンスを管理する本発明のコンテンツ配信システムと、各種のコンテンツを利用するデバイスのカテゴリが対応づけられるので、複数の DNK を同一のデバイスに持たせることができる。その結果、異なるカテゴリのコンテンツを 1 つのデバイスで管理することが可能となる。

図 2 1 は、この関係を表している。すなわち、デバイス D 1 には、コンテンツ配信システムに基づいて、DNK 1 が割り当てられている、コンテンツ 1 を利用するライセンス及びサービスデータが記録される。同様に、このデバイス D 1 には、例えば、DNK 2 が割り当てられた、メモリスティックに CD からリッピングしたコンテンツ 2 を記録することができる。この場合、デバイス D 1 は、コンテンツ 1 とコンテンツ 2 という、異なるシステム（コンテンツ配信システムとデバイス管理システム）により配信されたコンテンツを同時に扱うことが可能となる。新たな DNK を割り当てるとき、既に割り当てられている DNK を削除するなどして、デバイスに 1 個の DNK だけに対応させるようにした場合、このようなことはできない。

また、図 1 3 における、例えば、下側の 3 2 階層の各三角形の 1 つ 1 つに、図 2 2 に示されるライセンスカテゴリ 1 とライセンスカテゴリ 2 を割り当てることにより、同一のカテゴリ内を、サブカテゴリを利用して、コンテンツのジャンル、レーベル、販売店、配信サービス、コンテンツの出所、提供方法等の小さな集まりに分類して、管理することが可能となる。

図 2 2 の例においては、例えば、ライセンスカテゴリ 1 は、ジャズのジャンルに属し、ライセンスカテゴリ 2 は、ロックのジャンルに属する。ライセンスカテ

ゴリ 1 には、ライセンス ID が 1 であるコンテンツ 1 とコンテンツ 2 を対応させ、それぞれユーザ 1 乃至ユーザ 3 に配布されている。ライセンスカテゴリ 2 は、ライセンス ID 2 のコンテンツ 3、コンテンツ 4、およびコンテンツ 5 が含まれ、それぞれユーザ 1 とユーザ 3 に提供されている。

5 このように、本発明においては、カテゴリ毎に独立したキー管理が可能になる。

また、DNK を、機器やメディアに予め埋め込むのではなく、ライセンスサーバ 4 により、登録処理を行う際に、各機器やメディアにダウンロードするようにすることで、ユーザによるキーの取得が可能なシステムを実現することができる。

この場合のクライアント 1 の登録処理について、図 2 3 を参照して説明する。

10 ステップ S 1 6 1 において、クライアント 1 の CPU 2 1 は通信部 2 9 を制御してライセンスサーバ 4 にサービスデータ要求を送信する。ライセンスサーバ 4 の CPU 2 1 は、ステップ S 1 6 5 において、通信部 2 9 を介して入力されるサービスデータ要求を受信すると、S 1 6 6 において、通信部 2 9 を介してユーザ情報要求をクライアント 1 に送信する。

15 クライアント 1 の CPU 2 1 は、ステップ S 1 6 2 において、通信部 2 9 を介してユーザ情報要求を受信すると、出力部 2 7 を制御しディスプレイなどにユーザ情報の入力を促すメッセージを表示させる。ユーザがキーボードなどを操作することにより、入力部 2 6 からユーザ本人の個人情報や決済情報等のユーザ情報を入力すると、S 1 6 3 においてクライアント 1 の CPU 2 1 は、入力されたユーザ情報を、通信部 2 9 を介してライセンスサーバ 4 に送信する。

20 ライセンスサーバ 4 の CPU 2 1 は、ステップ S 1 6 7 において、通信部 2 9 を介してユーザ情報を受信すると、ステップ S 1 6 8 において、そのライセンスサーバ 4 に割り当てられたカテゴリのノード以下のリーフのうち、まだ割り当てられていないリーフをクライアント 1 に割り当て、そのリーフからライセンスサーバ 4 に割り当てられたカテゴリのノードまでのパス上のノードに割り当てられたノードキーの組をデバイスノードキーとして生成し、生成されたデバイスノードキー、クライアント 1 に割り当てられたリーフのリーフ ID、クライアント 1

の秘密鍵、クライアント 1 の秘密鍵・公開鍵のペア、ライセンスサーバの公開鍵、及び各公開鍵の証明書をまとめてサービスデータとして生成し、S 1 6 9 において通信部 2 9 を介してクライアントに生成されたサービスデータを送信すると共に、ドライブ 3 0 を制御してユーザ情報をリーフ ID と対応付けてハードディスク等の記録メディアに記録させる。

クライアント 1 の CPU 2 1 は、ステップ S 1 6 4 において、通信部 2 9 を介してサービスデータを受信すると、暗号化復号部 2 4 を制御して受信したサービスデータを暗号化し、ドライブ 3 0 を制御してハードディスク等の記録メディアに記録させる。

- 10 以上のようにして、ライセンスサーバ 4 はクライアント 1 及びそのユーザを登録し、クライアント 1 は所望のコンテンツ配信サービスを利用するために必要な、デバイスノードキーを含むサービスデータを受け取ることができる。

コンテンツは、それが作成された後、どのような使われ方をされようとも、その使われ方に関わりなく、全ての用途において、使用可能であるのが望ましい。

- 15 例えば、異なるコンテンツ配信サービス、あるいはドメインの使用状況が異なる場合においても、同一のコンテンツが使えることが望ましい。本発明においては、このため、上述したように、各ユーザ（クライアント 1）に、認証局としてのライセンスサーバ 4 から秘密鍵と、それに対応する公開鍵の証明書

(certificates) が配布される。各ユーザは、その秘密鍵を用いて、署名

- 20 (signature) を作成し、コンテンツに付加して、コンテンツの真正さ (integrity) を保証し、かつコンテンツの改竄防止を図ることができる。

この場合の処理の例について、図 2 4 のフローチャートを参照して説明する。図 2 4 の処理は、ユーザが CD から再生したデータを記憶部 2 8 に記憶させるリップリング処理を説明するものである。

- 25 最初に、ステップ S 1 7 1 において、クライアント 1 の CPU 2 1 は、通信部 2 9 を介して入力される CD の再生データを記録データとして取り込む。ステップ S 1 7 2 において、CPU 2 1 は、ステップ S 1 7 1 の処理で取り込まれた

記録データにウォーターマークが含まれているか否かを判定する。このウォーターマークは、3ビットのコピー管理情報（CCI）と、1ビットのトリガ

（Trigger）とにより構成され、コンテンツのデータの中に埋め込まれている。

5 CPU 2 1 は、ウォーターマークが検出された場合には、ステップ S 1 7 3 に進み、そのウォーターマークを抽出する処理を実行する。ウォーターマークが存在しない場合には、ステップ S 1 7 3 の処理はスキップされる。

次に、ステップ S 1 7 4 において、CPU 2 1 は、コンテンツに対応して記録するヘッダのデータを作成する。このヘッダのデータは、コンテンツ ID、ライセンス ID、ライセンスを取得するためのアクセス先を表す URL、およびウォーター

10 ターマークに含まれていたコピー管理情報（CCI）と、トリガ（Trigger）により構成される。

次に、ステップ S 1 7 5 に進み、CPU 2 1 は、ステップ S 1 7 4 の処理で作成したヘッダのデータに基づいたデジタル署名を、自分自身の秘密鍵を用いて作成する。この秘密鍵は、ライセンスサーバ 4 から取得したものである（図 7 の

15 テップ S 6 7）。

ステップ S 1 7 6 で、CPU 2 1 は、暗号化復号部 2 4 を制御し、コンテンツキーでコンテンツを暗号化させる。コンテンツキーは、乱数等を用いて生成される。

次に、ステップ S 1 7 7 において、CPU 2 1 は、ファイルフォーマットに基づき、データを、例えば、ミニディスク等により構成される光磁気ディスク 4 3

20 に記録させる。

なお、記録媒体がミニディスクである場合、ステップ S 1 7 6 において、CPU 2 1 は、コンテンツをコーデック部 2 5 に供給し、例えば、ATRAC 3 方式によりコンテンツを符号化させる。そして、符号化されたデータが暗号化復号部

25 2 4 によりさらに暗号化される。

図 2 5 は、以上のようにして、記録媒体にコンテンツが記録された状態を模式的に表している。暗号化されているコンテンツ（E（At 3））から抽出されたウォーターマーク（WM）が、コンテンツの外（ヘッダ）に記録されている。

図 2 6 は、コンテンツを記録媒体に記録する場合のファイルフォーマットのより詳細な構成を表している。この例においては、コンテンツ ID（CID）、ライセンス ID（LID）、URL、およびウォーターマーク（WM）を含むヘッダが記録されている他、EKB、コンテンツキー Kc をルートキー KR で暗号化したデータ（Enc（KR, Kc））、証明書（Cert）、ヘッダに基づき生成されたデジタル署名（Sig（Header））、コンテンツをコンテンツキー Kc で暗号化したデータ（Enc（Kc, Content））、メタデータ（Meta Data）およびマーク（Mark）が記録されている。

ウォーターマークは、コンテンツの内部に埋め込まれているものであるが、図 2 5 と図 2 6 に示されるように、コンテンツの内部とは別に、ヘッダ内に配置するようにすることで、ウォーターマークとしてコンテンツに埋め込まれている情報を迅速に、かつ簡単に検出することが可能となる。従って、そのコンテンツを、コピーすることができるか否かを、迅速に判定することができる。

なお、メタデータは、例えば、ジャケット、写真、歌詞等のデータを表す。マークについては、図 3 2 を参照して後述する。

図 2 7 は、証明書としての公開鍵証明書の例を表している。公開鍵証明書は、通常、公開鍵暗号方式における認証局（CA : Certificate Authority）が発行する証明書であり、ユーザが、認証局に提出した自己の ID や公開鍵などに、認証局が有効期限等の情報を付加し、さらに、認証局によるデジタル署名を付加して作成される。この発明においては、ライセンスサーバ 4（またはコンテンツサーバ 3）が、証明書と秘密鍵、従って公開鍵も発行するので、ユーザは、ユーザ ID、パスワード等をライセンスサーバ 4 に提供し登録処理を行うことによって、この公開鍵証明書を得ることができる。

図 2 7 における公開鍵証明書は、証明書のバージョン番号、ライセンスサーバ 4 が証明書の利用者（ユーザ）に対して割りつける証明書の通し番号、デジタル署名に用いたアルゴリズム、およびパラメータ、認証局（ライセンスサーバ 4）の名前、証明書の有効期限、証明書利用者の ID（ノード ID またはリーフ ID）、並びに証明書利用者の公開鍵が、メッセージとして含まれている。さらに、このメッセージには、認証局としてのライセンスサーバ 4 により作成されたデジタル署名が付加されている。このデジタル署名は、メッセージに対してハッシュ関数を適用して生成されたハッシュ値に基づいて、ライセンスサーバ 4 の秘密鍵を用いて生成されたデータである。

10 ノード ID またはリーフ ID は、例えば、図 1 2 の例の場合、デバイス 0 であれば「0 0 0 0」とされ、デバイス 1 であれば「0 0 0 1」とされ、デバイス 1 5 であれば「1 1 1 1」とされる。このような ID に基づいて、そのデバイス（エンティティ）がツリー構成のどの位置（リーフまたはノード）に位置するエンティティであるのかが識別される。

15 このように、コンテンツを利用するのに必要なライセンスを、コンテンツとは分離して配布するようにすることにより、コンテンツの配布が自由に行われることになる。任意の方法、あるいは経路で入手されたコンテンツは、一元的に取り扱うことが可能である。

また、ファイルフォーマットを図 2 6 に示されるように構成することで、その
20 フォーマットのコンテンツを、インターネットを介して配信する場合は勿論、SDMI（Secure Digital Music Initiative）機器に提供する場合においても、コンテンツの著作権を管理することが可能となる。

さらに、例えば、図 2 8 に示されるように、コンテンツが記録媒体を介して提供されたとしても、インターネット 2 を介して提供されたとしても、同様の処理
25 により、SDMI（Secure Digital Music Initiative）機器としての所定の PD（Portable Device）等に、チェックアウトしたりすることが可能となる。

次に、図 2 9 のフローチャートを参照して、クライアント 1 が他のクライアント（例えば、PD）に対してコンテンツをチェックアウトする場合の処理について説明する。

最初に、ステップ S 1 9 1 において、CPU 2 1 は、コンテンツにデジタル署名が付加されているか否かを判定する。デジタル署名が付加されていると判定された場合、ステップ S 1 9 2 に進み、CPU 2 1 は、証明書を抽出し、認証局（ライセンスサーバ 4）の公開鍵で検証する処理を実行する。すなわち、クライアント 1 は、ライセンスサーバ 4 からライセンスサーバ 4 の秘密鍵に対応する公開鍵を取得し、その公開鍵で公開鍵証明書に付加されているデジタル署名を復号する。図 2 7 を参照して説明したように、デジタル署名は、認証局（ライセンスサーバ 4）の秘密鍵に基づいて生成されており、ライセンスサーバ 4 の公開鍵を用いて復号することができる。さらに、CPU 2 1 は、証明書のメッセージ全体に対してハッシュ関数を適用してハッシュ値を演算する。そして CPU 2 1 は、演算されたハッシュ値と、デジタル署名を復号して得られたハッシュ値とを比較し、両者が一致すれば、メッセージは改竄されたものではないと判定する。両者が一致しない場合には、この証明書は、改竄されたものであるということになる。

そこで、ステップ S 1 9 3 において、CPU 2 1 は、証明書が改竄されていないか否かを判定し、改竄されていないと判定された場合、ステップ S 1 9 4 に進み、証明書を EKB で検証する処理を実行する。この検証処理は、証明書に含まれるリーフ ID（図 2 7）に基づいて、EKB をたどることができるか否かを調べることにより行われる。この検証について、図 3 0 と図 3 1 を参照して説明する。

いま、図 3 0 に示されるように、例えば、リーフキー K 1 0 0 1 を有するデバイスがリボークされたデバイスであるとする。このとき、図 3 1 に示されるようなデータ（暗号化キー）とタグを有する EKB が、各デバイス（リーフ）に配布される。この EKB は、図 3 0 におけるデバイス「1 0 0 1」をリボークするために、キー KR, K 1, K 1 0, K 1 0 0 を更新する EKB となっている。

リボークデバイス「1001」以外の全てのリーフは、更新されたルートキー $K(t)R$ を取得することができる。すなわち、ノードキー $K0$ の下位に連なるリーフは、更新されていないノードキー $K0$ を、デバイス内に保持しているので、暗号化キー $Enc(K0, K(t)R)$ を、キー $K0$ によって復号することで、更新
5 ルートキー $K(t)R$ を取得することができる。

また、ノード11以下のリーフは、更新されていないノードキー $K11$ を用いて、 $Enc(K11, K(t)1)$ をノードキー $K11$ によって復号することで、更新ノードキー $K(t)1$ を取得することができる。さらに、 $Enc(K(t)1, K(t)R)$ をノードキー $K(t)1$ によって復号することで、更新ルートキー
10 $K(t)R$ を取得することが可能となる。ノードキー $K101$ の下位リーフについても、同様に更新ルートキー $K(t)R$ を取得することが可能である。

さらに、リボークされていないリーフキー $K1000$ を有するデバイス「1000」は、自己のリーフキー $K1000$ で $Enc(K1000, K(t)1000)$ を復号して、ノードキー $K(t)100$ を取得することができ、これを用いてさらに、上位のノードキーを順次復号し、更新ルートキー $K(t)R$ を取得す
15 ることができる。

これに対して、リボークされたデバイス「1001」は、自己のリーフの1段上の更新ノードキー $K(t)100$ を、EKB 処理により取得できないので、結局、更新ルートキー $K(t)R$ を取得することができない。

20 リボークされていない正当なデバイス（クライアント1）には、図31に示されるデータとタグを有する EKB が、ライセンスサーバ4から配信され、格納されている。

そこで、各クライアントは、そのタグを利用して、EKB 追跡処理を行うことができる。この EKB 追跡処理は、上位のルートキーからキー配信ツリーをたど
25 れるか否かを判定する処理である。

例えば、図30のリーフ「1001」のID（リーフID）である「1001」を、「1」「0」「0」「1」の4ビットとして把握し、最上位ビットから順次、

下位ビットに従って、ツリーをたどることができるか否かが判定される。この判定では、ビットが1であれば、右側に進み、0であれば、左側に進む処理が行われる。

ID「1001」の最上位ビットが1であるから、図30のルートキーKRから
5 右側に進む。EKBの最初のタグ（番号0のタグ）は、0：{0, 0}であり、
両枝にデータを有するものであると判定される。この場合、右側に進むことができるので、ノードキーK1にたどり着くことができる。

次に、ノードキーK1の下位のノードに進む。ID「1001」の2番目のビットは0であるから左側に進む。番号1のタグは、左側のノードキーK0の下位
10 のデータの有無を表すものであり、ノードキーK1の下位のデータの有無を示す
タグは、番号2のタグである。このタグは、図31に示されるように、2：{0,
0}であり、両枝にデータを有するものとされる。従って、左側に進み、ノード
キーK10にたどり着くことができる。

さらに、ID「1001」の3番目のビットは0であり、左側に進む。このとき、
15 K10の下位のデータの有無を示すタグ（番号3のタグ）は、3：{0,
0}であり、両枝にデータを有するものと判定される。そこで、左側に進み、ノード
キーK100にたどり着くことができる。

さらに、ID「1001」の最下位ビットは1であり、右側に進む。番号4の
タグは、ノードキーK11に対応するものであり、K100の下位のデータの符
20 号を表すタグは、番号5のタグである。このタグは、5：{0, 1}である。従
って、右側には、データが存在しないことになる。その結果、ノード「100
1」にはたどり着けないことになり、ID「1001」のデバイスは、EKBによる
更新ルートキーを取得できないデバイス、すなわちリボークデバイスであると
判定される。

25 これに対して、例えば、リーフキーK1000を有するデバイスIDは、「1
000」であり、上述した場合と同様に、EKB内のタグに基づくEKB追跡処

理を行うと、ノード「1000」にたどり着くことができる。従って、ID「1000」のデバイスは、正当なデバイスであると判定される。

図29に戻って、CPU21は、ステップS194の検証処理に基づき、証明書がリボークされていないか否かをステップS195で判定し、証明書がリボークされていない場合には、ステップS196に進み、デジタル署名を証明書に含まれる公開鍵で検証する処理を実行する。

すなわち、図27に示されるように、証明書には、証明書利用者（コンテンツ作成者）の公開鍵が含まれており、この公開鍵を用いて、図26に示される署名（Sig（Header））が検証される。すなわち、この公開鍵を用いて、デジタル署名 Sig（Header）を復号して得られたデータ（ハッシュ値）と、図26に示される Header にハッシュ関数を適用して演算されたハッシュ値とを比較することで、両者が一致していれば、Header が改竄されていないことを確認することができる。これに対して、両者が一致しなければ、Header は改竄されているということになる。

ステップS197において、CPU21は、Header が改竄されているか否かを判定し、改竄されていない場合は、ステップS198に進み、ウォーターマークを検証する。ステップS199において、CPU21は、ウォーターマークの検証の結果、チェックアウトが可能であるか否かを判定する。チェックアウトが可能である場合には、ステップS200に進み、CPU21は、チェックアウトを実行する。すなわち、チェックアウト先のクライアント1に対してコンテンツを転送し、コピーさせる。

ステップS191において、デジタル署名が存在しないと判定された場合、ステップS193において、証明書が改竄されていると判定された場合、ステップS195において、証明書を EKB で検証することができなかったと判定された場合、ステップS197において、デジタル署名の検証の結果、ヘッダが改竄されていると判定された場合、または、ステップS199において、ウォーターマークにチェックアウトの禁止が記述されていると判定された場合、ステップS2

0 1に進み、エラー処理が実行される。すなわち、この場合には、チェックアウトが禁止される。

このように、証明書と秘密鍵をライセンスサーバ4からユーザに配布し、コンテンツ作成時に、デジタル署名を付加することにより、コンテンツの作成者の真正を保証することが可能となる。これにより、不正なコンテンツの流通を抑制することができる。

さらに、ウォーターマークをコンテンツ作成時に検出し、その情報をデジタル署名に付することで、ウォーターマーク情報の改竄を防止し、コンテンツの真正を保証することができる。

10 その結果、一度作成されたコンテンツは、どのような形態で配信されたとしても、元のコンテンツの真正を保証することが可能となる。

さらに、コンテンツは、使用条件を有さず、使用条件は、ライセンスに付加されているので、ライセンス内の使用条件を変更することで、それに関係するコンテンツの使用条件を一斉に変更することが可能となる。

15 次に、マークの利用方法について説明する。本発明においては、上述したように、使用条件は、コンテンツではなく、ライセンスに付加される。しかしながら、コンテンツによって、使用状況が異なる場合がある。そこで、本発明においては、図26に示されるように、コンテンツにマークが付加される。

ライセンスとコンテンツは、1対多の関係にあるため、コンテンツの個々の使用状況をライセンスの使用条件にのみ記述するのは困難となる。そこで、このように、コンテンツに使用状況を付加することにより、ライセンスでの管理をしながらも、個々のコンテンツを管理することが可能となる。

このマークには、例えば、図32に示されるように、ユーザのID（リーフID）、所有権フラグ、使用開始時刻、およびコピー回数等が記述される。

25 さらに、マークには、リーフID、所有権フラグ、使用開始時刻、およびコピー回数等のメッセージに基づいて生成されたデジタル署名が付加される。

所有権フラグは、例えば、所定の期間だけコンテンツを使用可能とするライセンスを、そのまま買い取ったような場合（使用期間を永久に変更したような場合）に付加される。使用開始時刻は、コンテンツの使用を所定の期間内に開始した場合に記述される。例えば、コンテンツをダウンロードする時期が制限されているような場合において、その期限内にダウンロードが行われたようなとき、その実際にコンテンツをダウンロードした日時がここに記述される。これにより、期間内での有効な使用であることが、証明される。

コピー回数には、それまでにそのコンテンツをコピーした回数が履歴（ログ）として記述される。

10 次に、図 3 3 のフローチャートを参照して、ユーザがライセンスを買い取った場合に、マークを付加する処理について、マークをコンテンツに付加する例として説明する。

最初に、ステップ S 2 2 1 において、CPU 2 1 は、入力部 2 6 からのユーザの指令に基づいて、インターネット 2 を介して、ライセンスサーバ 4 にアクセスする。

ステップ S 2 2 2 において、CPU 2 1 は、ユーザからの入力部 2 6 を介しての入力を取り込み、その入力に対応してライセンスサーバ 4 に対してライセンスの買い取りを要求する。

この要求に対応して、図 3 4 のフローチャートを参照して後述するように、ライセンスサーバ 4 は、ライセンスを買い取るために必要な対価を提示してくる（図 3 4 のステップ S 2 4 2）。そこで、ステップ S 2 2 3 において、クライアント 1 の CPU 2 1 は、ライセンスサーバ 4 からの対価の提示を受け取ると、これを出力部 2 7 に出力し、表示させる。

ユーザは、この表示に基づいて、提示された対価を了承するか否かを判断し、その判断結果に基づいて、入力部 2 6 からその判断結果を入力する。

CPU 2 1 は、ステップ S 2 2 4 において、入力部 2 6 からの入力に基づいて、ユーザが提示された対価を了承したか否かを判定し、了承したと判定した場合に

は、ステップ S 2 2 5 に進み、ライセンスサーバ 4 に了承を通知する処理を実行する。

この了承通知を受信すると、ライセンスサーバ 4 は、対価の買い取りを表す情報、すなわち所有権フラグを記述したマークを送信してくる（図 3 4 のステップ S 2 4 4）。そこで、ステップ S 2 2 6 において、クライアント 1 の CPU 2 1 は、ライセンスサーバ 4 からのマークを受け取ると、ステップ S 2 2 7 において、受け取ったマークをコンテンツに埋め込む処理を実行する。すなわち、これにより、買い取られたライセンスに対応するコンテンツのマークとして、図 3 2 に示されるような所有権フラグが記述されたマークがコンテンツに対応して記録されることになる。また、このとき、CPU 2 1 は、メッセージが更新されたことになるので、デジタル署名（図 2 6）も更新し、記録媒体に記録する。

ステップ S 2 2 4 において、ライセンスサーバ 4 から提示された対価が了承されていないと判定された場合、ステップ S 2 2 8 に進み、CPU 2 1 は、提示された対価を了承しないことをライセンスサーバ 4 に通知する。

15 このようなクライアント 1 の処理に対応して、ライセンスサーバ 4 は、図 3 4 のフローチャートに示す処理を実行する。

すなわち、最初に、ステップ S 2 4 1 において、ライセンスサーバ 4 の CPU 2 1 は、クライアント 1 からライセンス買い取りの要求が送信されてくると（図 3 3 のステップ S 2 2 2）、これを受け取り、ステップ S 2 4 2 において、対象とされているライセンスの買い取りに必要な対価を記憶部 2 8 から読み出し、これをクライアント 1 に送信する。

上述したように、このようにして提示された対価に対して、クライアント 1 から提示された対価を了承するか否かの通知が送信されてくる。

そこで、ステップ S 2 4 3 において、ライセンスサーバ 4 の CPU 2 1 は、クライアント 1 から了承通知を受信したか否かを判定し、了承通知を受信したと判定した場合、ステップ S 2 4 4 に進み、対象とされるライセンスの買い取りを表すメッセージを含むマークを生成し、自分自身の秘密鍵で、デジタル署名を付加

して、クライアント 1 に送信する。このようにして送信されたマークは、上述したように、クライアント 1 の記憶部 28 において、対応するコンテンツに記録される（図 33 のステップ S 227）。

5 ステップ S 243 において、クライアント 1 から了承通知が受信されていないと判定された場合、ステップ S 244 の処理はスキップされる。すなわち、この場合には、ライセンスの買い取り処理が最終的に行われなかったことになるので、マークは送信されない。

10 図 35 は、ステップ S 244 において、ライセンスサーバ 4 からクライアント 1 に対して送信されるマークの構成例を表している。この例においては、そのユーザのリーフ ID、所有権フラグ (Own)、並びにリーフ ID と所有権フラグを、ライセンスサーバ 4 の秘密鍵 S に基づいて生成されたデジタル署名 Sigs (LeafID, Own) により、マークが構成されている。

15 なお、このマークは、特定のユーザの特定のコンテンツに対してのみ有効なものである。対象とされるコンテンツがコピーされた場合には、そのコピーされたコンテンツに付随するマークは無効とされる。

このようにして、コンテンツとライセンスを分離し、使用条件をライセンスに対応させる場合においても、個々のコンテンツの使用状況に応じたサービスを実現することが可能となる。

20 次に、グルーピングについて説明する。複数の機器やメディアを適当に集め、その 1 つの集合内においては、コンテンツを自由に授受することができるようにすることは、グルーピングと称される。通常、このグルーピングは、個人の所有する機器やメディアにおいて行われる。このグルーピングは、従来、グループ毎にグループキーを設定する等して行われていたが、グループ化する複数の機器やメディアに、同一のライセンスを対応づけることにより、容易にグルーピングす
25 ることが可能となる。

また、各機器を予め登録しておくことで、グルーピングすることも可能である。この場合のグルーピングについて、以下に説明する。

この場合、ユーザは、グルーピング対象とされる機器の証明書を予めサーバに登録しておく必要がある。この証明書の登録処理について、図 3 6 と図 3 7 のフローチャートを参照して説明する。

最初に、図 3 6 のフローチャートを参照して、クライアント（グルーピング対象となる機器）の証明書の登録処理について説明する。ステップ S 2 6 1 において、クライアント 1 の CPU 2 1 は、グルーピングの対象とされる機器としての自分自身の証明書を作成する。この証明書には、自分自身の公開鍵が含まれる。

次に、ステップ S 2 6 2 に進み、CPU 2 1 は、ユーザの入力部 2 6 からの入力に基づいて、コンテンツサーバ 3 にアクセスし、ステップ S 2 6 3 において、ステップ S 2 6 1 の処理で作成された証明書をコンテンツサーバ 3 に送信する処理を実行する。

なお、証明書としては、ライセンスサーバ 4 から受信したものを、そのまま使用することもできる。

以上の処理は、グルーピング対象とされる全ての機器が行う。

次に、図 3 7 のフローチャートを参照して、図 3 6 のクライアント 1 の証明書の登録処理に対応して行われるコンテンツサーバ 3 の証明書の登録処理について説明する。

最初に、ステップ S 2 7 1 において、コンテンツサーバ 3 の CPU 2 1 は、クライアント 1 から送信されてきた証明書を受信すると、ステップ S 2 7 2 において、その証明書を記憶部 2 8 に登録する。

以上の処理が、グループ対象とされる機器毎に行われる。その結果、コンテンツサーバ 3 の記憶部 2 8 には、例えば、図 3 8 に示されるように、グループ毎に、そのグループを構成するデバイスの証明書が登録される。

図 3 8 に示される例では、グループ 1 の証明書として、証明書 C 1 1 乃至 C 1 4 が登録されている。これらの証明書 C 1 1 乃至 C 1 4 には、対応する公開鍵 K_{P11} 乃至 K_{P14} が含まれている。

同様に、グループ 2 の証明書として、証明書 C 2 1 乃至 C 2 3 が登録されており、これら是对応する公開鍵 K_{P21} 乃至 K_{P23} が含まれている。

以上のようなグループを構成する各機器毎に、その証明書が登録された状態において、ユーザからそのグループに属する機器にコンテンツの提供が要求されると、コンテンツサーバ 3 は、図 3 9 のフローチャートに示す処理を実行する。

最初に、ステップ S 2 8 1 において、コンテンツサーバ 3 の CPU 2 1 は、記憶部 2 8 に記憶されている証明書のうち、そのグループに属する証明書を検証する処理を実行する。

この検証処理は、図 3 0 と図 3 1 を参照して説明されたように、各機器の証明書に含まれるリーフ ID に基づいて、タグを利用して EKB をたどることで行われる。EKB は、コンテンツサーバ 3 にも、ライセンスサーバ 4 から配布されている。この検証処理により、リボークされている証明書は除外される。

ステップ S 2 8 2 において、コンテンツサーバ 3 の CPU 2 1 は、ステップ S 2 8 1 の検証処理の結果、有効とされた証明書を選択する。そして、ステップ S 2 8 3 において、CPU 2 1 は、ステップ S 2 8 2 の処理で選択された各機器の証明書の各公開鍵でコンテンツ鍵を暗号化する。ステップ S 2 8 4 において、CPU 2 1 は、対象とされるグループの各機器に、ステップ S 2 8 3 の処理で暗号化されたコンテンツ鍵をコンテンツとともに送信する。

図 3 8 に示されるグループ 1 のうち、例えば、証明書 C 1 4 がリボークされているとすると、ステップ S 2 8 3 の処理で、例えば、図 4 0 に示されるような暗号化データが生成される。

すなわち、図 4 0 の例においては、コンテンツ鍵 K_c が、証明書 C 1 1 の公開鍵 K_{P11} 、証明書 C 1 2 の公開鍵 K_{P12} 、または証明書 C 1 3 の公開鍵 K_{P13} により、暗号化されている。

コンテンツサーバ 3 の図 3 9 に示されるような処理に対応して、コンテンツの提供を受ける各グループの機器（クライアント）は、図 4 1 のフローチャートに示す処理を実行する。

最初に、ステップ S 2 9 1 において、クライアント 1 の CPU 2 1 は、コンテンツサーバ 3 が図 3 9 のステップ S 2 8 4 の処理で送信してきたコンテンツを、コンテンツ鍵とともに受信する。コンテンツは、コンテンツ鍵 Kc により、暗号化されており、コンテンツ鍵は上述したように、各機器が保持する公開鍵により
5 暗号化されている（図 4 0）。

そこで、ステップ S 2 9 2 において、CPU 2 1 は、ステップ S 2 9 1 の処理で受信した自分宛のコンテンツ鍵を、自分自身の秘密鍵で復号し、取得する。そして、取得したコンテンツ鍵を用いてコンテンツの復号処理が行われる。

例えば、図 4 0 の例に示される証明書 C 1 1 に対応する機器は、公開鍵 K_{P11} に対応する自分自身の秘密鍵を用いて、コンテンツ鍵 Kc の暗号を復号し、コンテンツ鍵 Kc を取得する。そして、コンテンツ鍵 Kc を用いて、コンテンツがさらに復号される。
10

同様の処理は、証明書 C 1 2, C 1 3 に対応する機器においても行われる。リボークされている証明書 C 1 4 の機器は、自分自身の公開鍵を用いて暗号化されたコンテンツ鍵 Kc がコンテンツに付随して送られてこないで、コンテンツ鍵 Kc を復号することができず、従って、コンテンツ鍵 Kc を用いてコンテンツを復号することができない。
15

以上においては、コンテンツキー（すなわちコンテンツ）に対してグルーピングを行うようにしたが、ライセンスキー（ライセンス）に対してグルーピングを行うことも可能である。
20

以上のようにして、特別なグループキーや、後述する ICV（Integrity Check Value）を用いずにグループ化が可能となる。このグループ化は、小規模のグループに適用するのに向いている。

本発明においては、ライセンスもチェックアウト、あるいはチェックインしたり、ムーブしたり、コピーしたりすることが可能とされる。但し、これらの処理は SDMI で定められたルールに基づいて行われる。
25

次に、図 4 2 と図 4 3 のフローチャートを参照して、このようなクライアントによるライセンスのチェックアウト処理について説明する。

最初に、図 4 2 のフローチャートを参照して他のクライアントにライセンスをチェックアウトするクライアントの処理について説明する。最初に、ステップ S 3 0 1 において、クライアント 1 の CPU 2 1 は、チェックアウト対象のライセンスのチェックアウト回数 N 1 を読み取る。このチェックアウト回数は、図 8 に示される使用条件に書き込まれているので、この使用条件から読み取られる。

次に、ステップ S 3 0 2 において、CPU 2 1 は、チェックアウト対象のライセンスの最大チェックアウト回数 N 2 を、やはりライセンスの使用条件から読み取る。

そして、ステップ S 3 0 3 において、CPU 2 1 は、ステップ S 3 0 1 の処理で読み取られたチェックアウト回数 N 1 と、ステップ S 3 0 2 の処理で読み取られた最大チェックアウト回数 N 2 とを比較し、チェックアウト回数 N 1 が最大チェックアウト回数 N 2 より小さいか否かを判定する。

チェックアウト回数 N 1 が、最大チェックアウト回数 N 2 より小さいと判定された場合、ステップ S 3 0 4 に進み、CPU 2 1 は、相手側の装置（チェックアウト先のクライアント）のリーフキーを相手個々の装置から取得し、そのリーフキーを、いまチェックアウト対象とされているライセンス ID に対応して記憶部 2 8 のチェックアウトリストに記憶させる。

次に、ステップ S 3 0 5 において、CPU 2 1 は、ステップ S 3 0 1 の処理で読み取られたライセンスのチェックアウト回数 N 1 の値を 1 だけインクリメントする。ステップ S 3 0 6 において、CPU 2 1 は、ライセンスのメッセージに基づいて、ICV を演算する。この ICV については、図 4 7 乃至図 5 1 を参照して後述する。ICV を用いてライセンスの改竄を防止することが可能となる。

次に、ステップ S 3 0 7 において、CPU 2 1 は、チェックアウト対象のライセンスと、ステップ S 3 0 6 の処理で演算された ICV を、自分自身の公開鍵を用いて暗号化して、EKB および証明書とともに、相手側の装置に出力し、コピ

一させる。さらに、ステップS 3 0 8において、CPU 2 1は、ステップS 3 0 6の処理で演算されたICVを、相手側装置のリーフキーと、ライセンスIDに対応して記憶部2 8のチェックリスト中に記憶させる。

5 ステップS 3 0 3において、チェックアウト回数N 1が最大チェックアウト回数N 2より小さくない（例えば、等しい）と判定された場合、もはや許容される回数だけチェックアウトが行われているので、これ以上チェックアウトを行うことができない。そこで、ステップS 3 0 9に進み、CPU 2 1は、エラー処理を実行する。すなわち、この場合、チェックアウト処理は実行されないことになる。

次に、図4 3のフローチャートを参照して、図4 2のチェックアウト処理により、ライセンスのチェックアウトを受けるクライアントの処理について説明する。

最初に、ステップS 3 2 1において、相手側装置（ライセンスをチェックアウトするクライアント1）に、自分自身のリーフキーを送信する。このリーフキーは、ステップS 3 0 4において、相手側のクライアントにより、ライセンスIDに対応して記憶される。

15 次に、ステップS 3 2 2において、CPU 2 1は、相手側のクライアント1から暗号化されたライセンスとICVが、EKBおよび証明書とともに送信されてきた場合、これを受信する。すなわち、このライセンス、ICV、EKBおよび証明書は、図4 2のステップS 3 0 7の処理で相手側の装置から送信されたものである。

20 ステップS 3 2 3において、CPU 2 1は、ステップS 3 2 2の処理で受信したライセンス、ICV、EKBおよび証明書を、記憶部2 8に記憶させる。

以上のようにして、ライセンスのチェックアウトを受けたクライアント1は、チェックアウトを受けたそのライセンスを使用して、所定のコンテンツを再生する場合、図4 4のフローチャートに示される処理を実行する。

25 すなわち、最初に、ステップS 3 4 1において、クライアント1のCPU 2 1は、ユーザより入力部2 6を介して再生が指定されたコンテンツのICVを演算する。そして、ステップS 3 4 2において、CPU 2 1は、記憶部2 8に記憶さ

れている暗号化されている ICV を、証明書に含まれている公開鍵に基づいて、復号させる。

- 次に、ステップ S 3 4 3 において、CPU 2 1 は、ステップ S 3 4 1 の処理により、いま演算された ICV と、ステップ S 3 4 2 の処理により読み出され、復号された ICV が一致するか否かを判定する。両者が一致する場合には、ライセンスは改竄されていないことになる。そこで、ステップ S 3 4 4 にすすみ、CPU 2 1 は、対応するコンテンツを再生する処理を実行する。

- これに対して、ステップ S 3 4 3 において、2 つの ICV が一致しないと判定された場合、ライセンスは改竄されている恐れがある。このため、ステップ S 3 4 5 に進み、CPU 2 1 は、エラー処理を実行する。すなわち、このとき、そのライセンスを用いてコンテンツを再生することができないことになる。

次に、以上のようにして、他のクライアントに一旦チェックアウトしたライセンスのチェックインを受けるクライアントの処理について、図 4 5 のフローチャートを参照して説明する。

- 最初に、ステップ S 3 6 1 において、CPU 2 1 は、相手側の装置（ライセンスを返却（チェックイン）してくるクライアント 1）のリーフキーと、チェックイン対象のライセンスの ID を取得する。次に、ステップ S 3 6 2 において、CPU 2 1 は、ステップ S 3 6 1 で取得されたチェックイン対象のライセンスが、自分自身が相手側装置にチェックアウトしたライセンスであるか否かを判定する。この判定は、図 4 2 のステップ S 3 0 8 の処理で記憶された ICV、リーフキー、およびライセンス ID に基づいて行われる。すなわち、ステップ S 3 6 1 で取得されたリーフキー、ライセンス ID、および ICV が、チェックアウトリスト中に記憶されているか否かが判定され、記憶されている場合には、自分自身がチェックアウトしたライセンスであると判定される。
- ライセンスが、自分自身がチェックアウトしたものであるとき、ステップ S 3 6 3 において、CPU 2 1 は、相手側の装置のライセンス、EKB および証明書の

削除を要求する。後述するように、この要求に基づいて、相手側の装置は、ライセンス、EKB および証明書の削除を実行する（図 4 6 のステップ S 3 8 3）。

ステップ S 3 6 4 において、CPU 2 1 は、一旦チェックアウトしたライセンスが再びチェックインされてきたので、そのライセンスのチェックアウト回数 N
5 1 を 1 だけデクリメントする。

ステップ S 3 6 5 において、CPU 2 1 は、相手側の装置に他のライセンスをチェックアウトしているか否かを判定し、まだチェックアウトしている他のライセンスが存在しない場合には、ステップ S 3 6 6 に進み、CPU 2 1 は、相手側の装置のチェックイン対象機器としてのチェックアウトリストにおける記憶を削
10 除する。これに対して、ステップ S 3 6 5 において、相手側の装置にチェックアウトしている他のライセンスが存在すると判定された場合には、他のライセンスのチェックインを受ける可能性があるので、ステップ S 3 6 6 の処理はスキップされる。

ステップ S 3 6 2 において、チェックイン対象とされているライセンスが、自
15 分自身が相手側装置にチェックアウトしたライセンスではないと判定された場合、CPU 2 1 は、ステップ S 3 6 7 に進み、エラー処理を実行する。すなわち、この場合には、自分自身が管轄するライセンスではないことになるので、チェックイン処理は実行されない。

ユーザが、ライセンスを不正にコピーしたような場合、記憶されている ICV
20 の値と、ステップ S 3 6 1 の処理で取得されたライセンスに基づいて演算された ICV の値が異なるものとなるで、チェックインできないことになる。

図 4 6 は、図 4 5 のフローチャートに示されるライセンスのチェックイン処理を実行するクライアントに対して、自分自身が有しているライセンスをチェックインさせるクライアントの処理を表している。

25 ステップ S 3 8 1 において、クライアント 1 の CPU 2 1 は、相手側の装置（図 4 5 のフローチャートに示す処理を実行するクライアント 1）にリーフキーとチェックイン対象のライセンスの ID を送信する。上述したように、相手側の

装置は、ステップS 3 6 1において、このリーフキーとライセンス ID を取得し、ステップS 3 6 2において、それに基づいて、チェックイン対象のライセンスの認証処理を実行する。

5 ステップS 3 8 2において、クライアント1のCPU 2 1は、相手側の装置からライセンスの削除を要求されたか否かを判定する。すなわち、ライセンスが正当なチェックイン対象のライセンスである場合、上述したように、相手側の装置は、ステップS 3 6 3の処理でライセンス、EKB および証明書の削除を要求してくる。そこで、この要求を受信した場合、ステップS 3 8 3に進み、CPU 2 1は、ライセンス、EKB および証明書を削除する。すなわち、これにより、このクライアント1は、以後そのライセンスを使用できない状態となり、図4 5の
10 ステップS 3 6 4の処理により、チェックアウト回数N 1が、1だけデクリメントされるので、チェックインが完了したことになる。

 ステップS 3 8 2において、相手側の装置からライセンスの削除が要求されていないと判定された場合、ステップS 3 8 4に進み、エラー処理が実行される。
15 すなわち、この場合には、ICV の値が異なっている等の理由により、チェックインができないことになる。

 以上においては、チェックインとチェックアウトについて説明したが、同様に、ライセンスをコピーあるいはムーブさせるようにすることも可能である。

 次に、ライセンス（コンテンツも同様）の改竄を防止するためにライセンスの
20 インテグリティ・チェック値（ICV）を生成して、ライセンスに対応付けて、ICVの計算により、ライセンス改竄の有無を判定する処理構成について説明する。

 ライセンスのインテグリティ・チェック値（ICV）は、例えばライセンスに対するハッシュ関数を用いて計算され、 $ICV = hash(K_{icv}, L1, L2, \dots)$ によって計算される。K i c vはICV生成キーである。L 1, L
25 2はライセンスの情報であり、ライセンスの重要情報のメッセージ認証符号（MAC : Message authentication Code）が使用される。

DES暗号処理構成を用いたMAC値生成例を図47に示す。図47の構成に示すように対象となるメッセージを8バイト単位に分割し、(以下、分割されたメッセージをM1、M2、・・・、MNとする)、まず、初期値(IV)とM1を、演算部24-1Aにより排他的論理和する(その結果をI1とする)。次に、

5 I1をDES暗号化部24-1Bに入れ、鍵(以下、K1とする)を用いて暗号化する(出力をE1とする)。続けて、E1およびM2を演算部24-2Aにより排他的論理和し、その出力I2をDES暗号化部24-2Bへ入れ、鍵K1を用いて暗号化する(出力E2)。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。DES暗号化部24-NBから最後に出てきたENがメ

10 ッセージ認証符号(MAC(Message Authentication Code))となる。

このようなライセンスのMAC値とICV生成キーにハッシュ関数を適用してライセンスのインテグリティ・チェック値(ICV)が生成される。例えばライセンス生成時に生成したICVと、新たにライセンスに基づいて生成したICVとを比較して同一のICVが得られればライセンスに改竄のないことが保証され、

15 ICVが異なれば、改竄があったと判定される。

次に、ライセンスのインテグリティ・チェック値(ICV)生成キーであるK_{icv}を上述の有効化キーブロックによって送付する構成について説明する。すなわちEKBによる暗号化メッセージデータをライセンスのインテグリティ・チェック値(ICV)生成キーとした例である。

20 図48および図49に複数のデバイスに共通のライセンスを送付した場合、それらのライセンスの改竄の有無を検証するためのインテグリティ・チェック値生成キーK_{icv}を有効化キーブロック(EKB)によって配信する構成例を示す。図48はデバイス0, 1, 2, 3に対して復号可能なチェック値生成キーK_{icv}を配信する例を示し、図49はデバイス0, 1, 2, 3中のデバイス3をリボーク(排除)してデバイス0, 1, 2に対してのみ復号可能なチェック値生成キーK_{icv}を配信する例を示す。

25

図48の例では、更新ノードキー $K(t)00$ によって、チェック値生成キー $Kicv$ を暗号化したデータ $Enc(K(t)00, Kicv)$ とともに、デバイス0, 1, 2, 3においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキー $K(t)00$ を復号可能な有効化キーブロック(EKB)を生成して配信する。それぞれのデバイスは、図48の右側に示すように、まず、 EKB を処理(復号)することにより、更新されたノードキー $K(t)00$ を取得し、次に、取得したノードキー $K(t)00$ を用いて、暗号化されたチェック値生成キー $Enc(K(t)00, Kicv)$ を復号して、チェック値生成キー $Kicv$ を得ることが可能となる。

- 10 その他のデバイス4, 5, 6, 7...は同一の有効化キーブロック(EKB)を受信しても自身の保有するノードキー、リーフキーでは、 EKB を処理して更新されたノードキー $K(t)00$ を取得することができないので、安全に正当なデバイスに対してのみチェック値生成キーを送付することができる。

- 一方、図49の例は、図12の点線枠で囲んだグループにおいてデバイス3が、
15 例えば鍵の漏洩によりリボーク(排除)されているとして、他のグループのメンバー、すなわち、デバイス0, 1, 2, に対してのみ復号可能な有効化キーブロック(EKB)を生成して配信した例である。図49に示す有効化キーブロック(EKB)と、チェック値生成キー($Kicv$)をノードキー($K(t)00$)で暗号化したデータ $Enc(K(t)00, Kicv)$ を配信する。

- 20 図49の右側には、復号手順を示してある。デバイス0, 1, 2は、まず、受信した有効化キーブロックから自身の保有するリーフキーまたはノードキーを用いた復号処理により、更新ノードキー($K(t)00$)を取得する。次に、 $K(t)00$ による復号によりチェック値生成キー $Kicv$ を取得する。

- 図12に示す他のグループのデバイス4, 5, 6...は、この同様のデータ
25 (EKB)を受信したとしても、自身の保有するリーフキー、ノードキーを用いて更新ノードキー($K(t)00$)を取得することができない。同様にリボークされたデバイス3においても、自身の保有するリーフキー、ノードキーでは、更

新ノードキー ($K(t) 00$) を取得することができず、正当な権利を有するデバイスのみがチェック値生成キーを復号して利用することが可能となる。

このように、EKBを利用したチェック値生成キーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能としたチェック値生成キーを配信することが可能となる。

このようなライセンスのインテグリティ・チェック値 (ICV) を用いることにより、EKBと暗号化ライセンスの不正コピーを排除することができる。例えば図50Aに示すように、ライセンスL1とライセンスL2とをそれぞれのライセンスキーを取得可能な有効化キーブロック (EKB) とともに格納したメディア1があり、これをそのままメディア2にコピーした場合を想定する。EKBと暗号化ライセンスのコピーは可能であり、これを、EKBを復号可能なデバイスでは利用できることになる。

図50Bに示す例では、各メディアに正当に格納されたライセンスに対応付けてインテグリティ・チェック値 (ICV (L1, L2)) を格納する構成とする。

なお、(ICV (L1, L2)) は、ライセンスL1とライセンスL2にハッシュ関数を用いて計算されるライセンスのインテグリティ・チェック値である $ICV = hash(K_{icv}, L1, L2)$ を示している。図50Bの構成において、メディア1には正当にライセンス1とライセンス2が格納され、ライセンスL1とライセンスL2に基づいて生成されたインテグリティ・チェック値 (ICV (L1, L2)) が格納される。また、メディア2には正当にライセンス1が格納され、ライセンスL1に基づいて生成されたインテグリティ・チェック値 (ICV (L1)) が格納される。

この構成において、メディア1に格納された {EKB, ライセンス2} をメディア2にコピーしたとすると、メディア2で、ライセンスチェック値を新たに生成すると、ICV (L1, L2) が生成されることになり、メディア2に格納されている $K_{icv} (L1)$ と異なり、ライセンスの改竄あるいは不正なコピーによる新たなライセンスの格納が実行されたことが明らかになる。メディアを再生

するデバイスにおいて、再生ステップの前ステップに I C V チェックを実行して、生成 I C V と格納 I C V の一致を判別し、一致しない場合は、再生を実行しない構成とすることにより、不正コピーのライセンスの再生を防止することが可能となる。

- 5 また、さらに、安全性を高めるため、ライセンスのインテグリティ・チェック値（I C V）を書き換えカウンタを含めたデータに基づいて生成する構成としてもよい。すなわち $I C V = \text{hash}(K i c v, \text{counter} + 1, L 1, L 2, \dots)$ によって計算する構成とする。ここで、カウンタ（ $\text{counter} + 1$ ）は、I C V の書き換えごとに 1 つインクリメントされる値として設定する。
- 10 なお、カウンタ値はセキュアなメモリに格納する構成とすることが必要である。

さらに、ライセンスのインテグリティ・チェック値（I C V）をライセンスと同一メディアに格納することができない構成においては、ライセンスのインテグリティ・チェック値（I C V）をライセンスとは別のメディア上に格納する構成としてもよい。

- 15 例えば、読み込み専用メディアや通常の MO 等のコピー防止策のとられていないメディアにライセンスを格納する場合、同一メディアにインテグリティ・チェック値（I C V）を格納すると I C V の書き換えが不正なユーザによりなされる可能性があり、I C V の安全性が保てないおそれがある。この様な場合、ホストマシン上の安全なメディアに I C V を格納して、ライセンスのコピーコントロール（例えば check-in/check-out、move）に I C V を使用する構成とすることにより、I C V の安全な管理およびライセンスの改竄チェックが可能となる。
- 20 この構成例を図 5 1 に示す。図 5 1 では読み込み専用メディアや通常の MO 等のコピー防止策のとられていないメディア 2 2 0 1 にライセンス 1 乃至ライセンス 3 が格納され、これらのライセンスに関するインテグリティ・チェック値（I

- 25 C V）を、ユーザが自由にアクセスすることの許可されないホストマシン上の安全なメディア 2 2 0 2 に格納し、ユーザによる不正なインテグリティ・チェック値（I C V）の書き換えを防止した例である。このような構成として、例えばメ

ディア 2 2 0 1 を装着したデバイスが、メディア 2 2 0 1 の再生を実行する際にホストマシンである P C、サーバにおいて I C V のチェックを実行して再生の可否を判定する構成とすれば、不正なコピーライセンスあるいは改竄ライセンスの再生を防止できる。

- 5 本発明が適用されるクライアントは、いわゆるパーソナルコンピュータ以外に、PDA (Personal Digital Assistants)、携帯電話機、ゲーム端末機などとしてすることができる。

一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、ネットワークや記録媒体からインストールされる。

10

この記録媒体は、図 2 に示されるように、装置本体とは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク 4 1 (フロッピーディスクを含む)、光ディスク 4 2 (CD-ROM(Compact Disk - Read Only Memory),DVD(Digital Versatile Disk)を含む)、光磁気ディスク 4 3 (MD (Mini-Disk) を含む)、もしくは半導体メモリ 4 4 などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに提供される、プログラムが記録されている ROM 2 2 や、記憶部 2 8 に含まれるハードディスクなどで構成される。

15

20

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

- 25 また、セキュリティに関連する処理を実行させるプログラムは、その処理を解析されるのを防ぐため、そのプログラム自体が暗号化されているのが望ましい。

例えば、暗号処理などを行う処理については、そのプログラムをタンパーレジスタントモジュールとして構成することができる。

- また、コンテンツを利用許可するライセンスを特定するためにコンテンツのヘッダに記載されている情報はライセンスを一意に識別するライセンス ID でなく
- 5 てもよい。上記の実施例では、ライセンス I.D が、コンテンツの利用に必要なライセンスを特定する情報であり、あるライセンスが利用を許可するコンテンツを特定する情報であり、クライアント 1 からライセンス要求によって要求されるライセンスを識別する情報である。コンテンツにコンテンツのそのコンテンツに関する各種属性情報のリストが記載され、ライセンスに、そのライセンスによって
- 10 利用許可されるコンテンツの条件式を記載するようにしても良い。この場合では、コンテンツに含まれる属性情報がそのコンテンツの利用を許可するライセンスを特定する情報であり、ライセンスに含まれる条件式がそのライセンスが利用を許可するコンテンツを特定する情報であり、ライセンス I D はライセンスを一意に識別する情報となる。このようにした場合には、一つのコンテンツに複数のライ
- 15 センスを対応付けることが可能になり、ライセンスの発行を柔軟に行うことができる。

また、コンテンツデータは音楽データに限らない。例えばコンテンツは、画像データ、動画データ、テキストデータ、アニメーションデータ、ソフトウェアプログラム、あるいはそれらを組み合わせたものであっても良い。

- 20 また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

- 本発明の情報処理装置によれば、ライセンスに関する使用状況情報に対応する
- 25 マーク情報をクライアントに提供するようにしたので、コンテンツとライセンスを分離し、コンテンツを比較的自由に流通させつつ、コンテンツを個別に管理することが可能となる。

請求の範囲

1. クライアントからのアクセスに基づいて、前記クライアントにコンテンツのライセンスを提供する情報処理装置において、

5 前記クライアントから前記ライセンスを指定する指定情報を取得する第1の取得手段と、

前記第1の取得手段により取得された前記指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得手段と、

前記第2の取得手段により取得された前記使用状況情報に対応するマーク情報を生成し、前記クライアントに提供する提供手段と

10 を備えることを特徴とする情報処理装置。

2. 前記マーク情報は、前記ライセンスを識別する識別情報、前記ライセンスの買い取りを表す情報、前記ライセンスが対象とするコンテンツの使用を開始した日時、または前記コンテンツをコピーした回数を含む

ことを特徴とする請求の範囲第1項に記載の情報処理装置。

15 3. クライアントからのアクセスに基づいて、前記クライアントにコンテンツのライセンスを提供する情報処理装置の情報処理方法において、

前記クライアントから前記ライセンスを指定する指定情報を取得する第1の取得ステップと、

20 前記第1の取得ステップの処理により取得された前記指定情報により指定されたライセンスに関する使用状況情報を取得する第2の取得ステップと、

前記第2の取得ステップの処理により取得された前記使用状況情報に対応するマーク情報を生成し、前記クライアントに提供する提供ステップと

を含むことを特徴とする情報処理方法。

25 4. クライアントからのアクセスに基づいて、前記クライアントにコンテンツのライセンスを提供する情報処理装置用のプログラムにおいて、

前記クライアントから前記ライセンスを指定する指定情報を取得する第1の取得ステップと、

前記第 1 の取得ステップの処理により取得された前記指定情報により指定されたライセンスに関する使用状況情報を取得する第 2 の取得ステップと、

前記第 2 の取得ステップの処理により取得された前記使用状況情報に対応するマーク情報を生成し、前記クライアントに提供する提供ステップと

- 5 を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

5. クライアントからのアクセスに基づいて、前記クライアントにコンテンツのライセンスを提供する情報処理装置を制御するコンピュータが実行可能なプログラムであって、

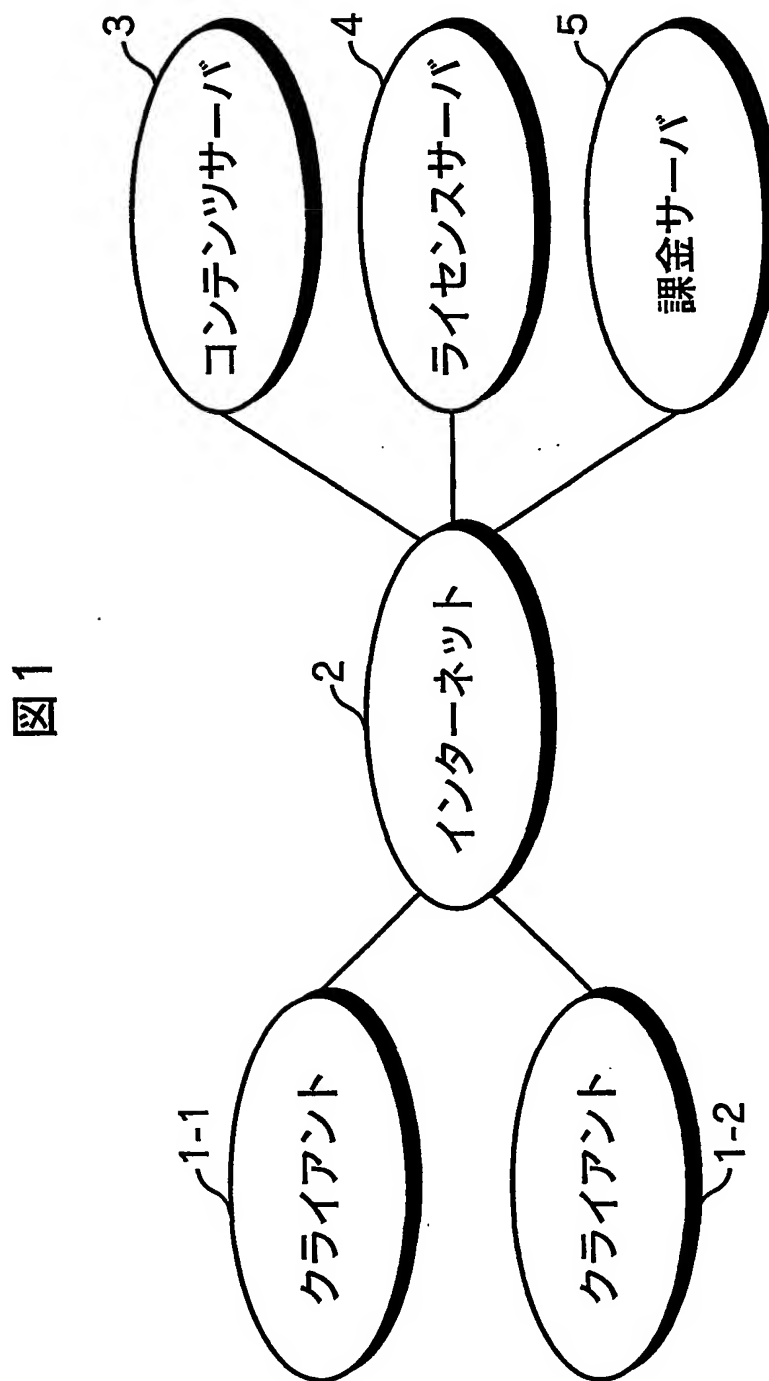
- 10 前記クライアントから前記ライセンスを指定する指定情報を取得する第 1 の取得ステップと、

前記第 1 の取得ステップの処理により取得された前記指定情報により指定されたライセンスに関する使用状況情報を取得する第 2 の取得ステップと、

前記第 2 の取得ステップの処理により取得された前記使用状況情報に対応する

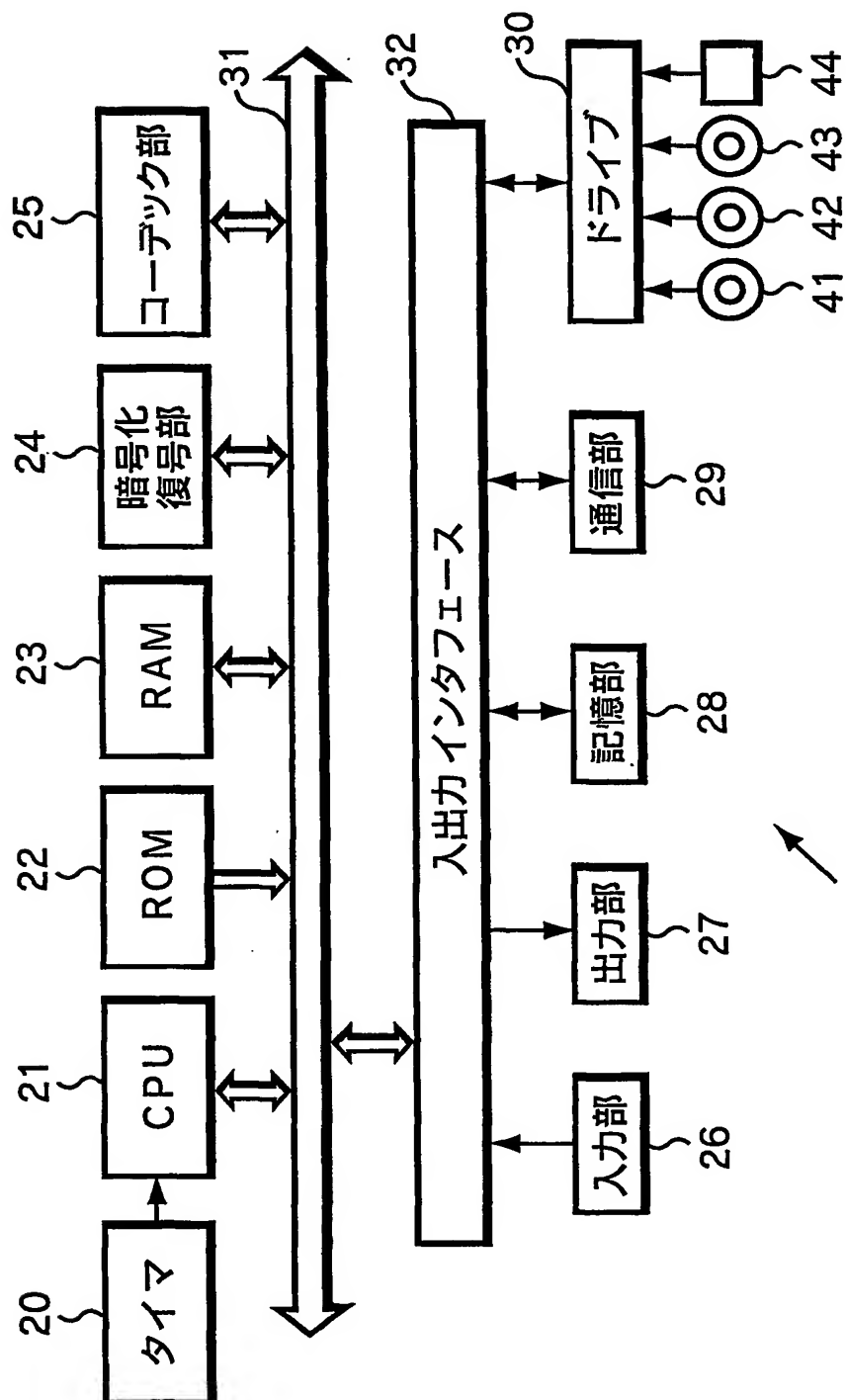
- 15 マーク情報を生成し、前記クライアントに提供する提供ステップと
 を含むことを特徴とするプログラム。

1/45



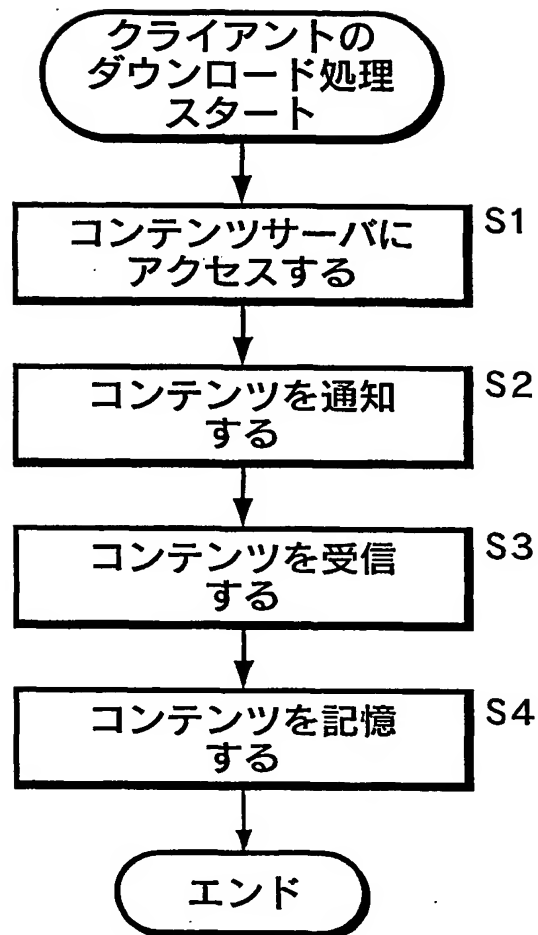
2/45

図2



3/45

図 3



4/45

図 4

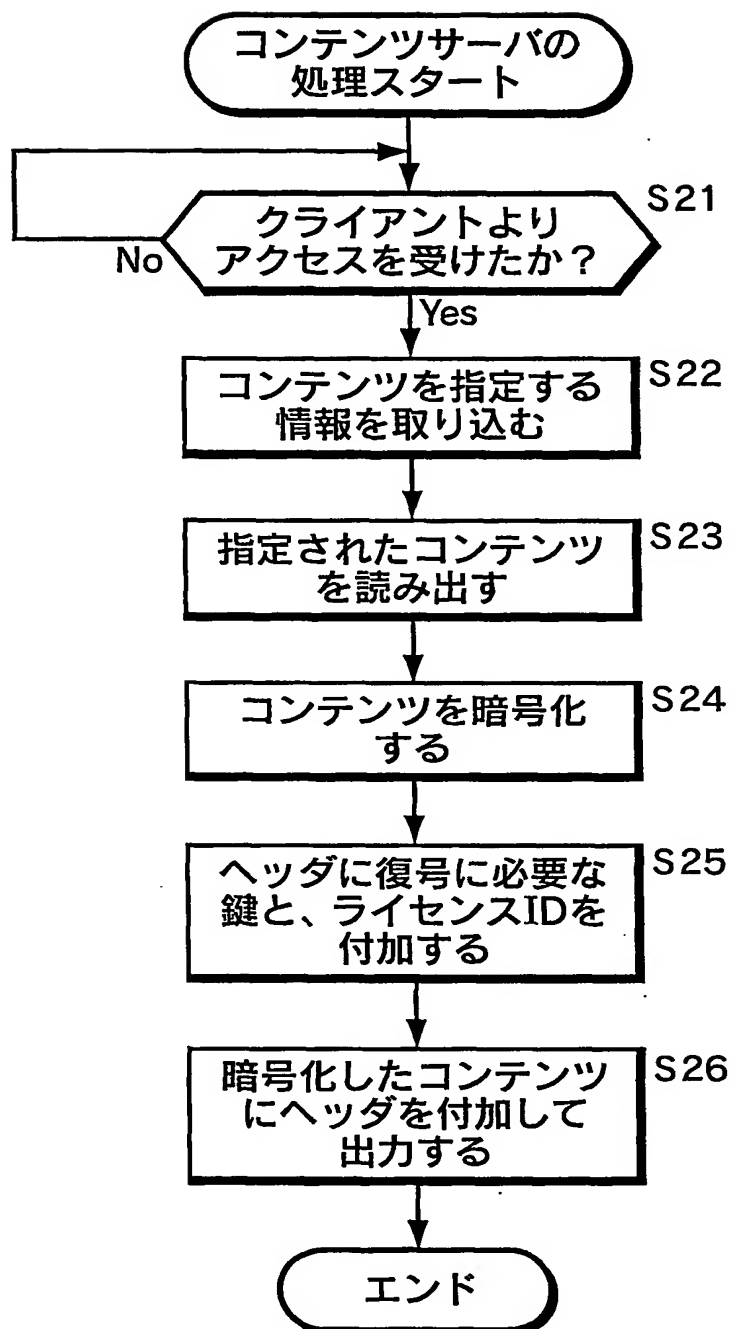
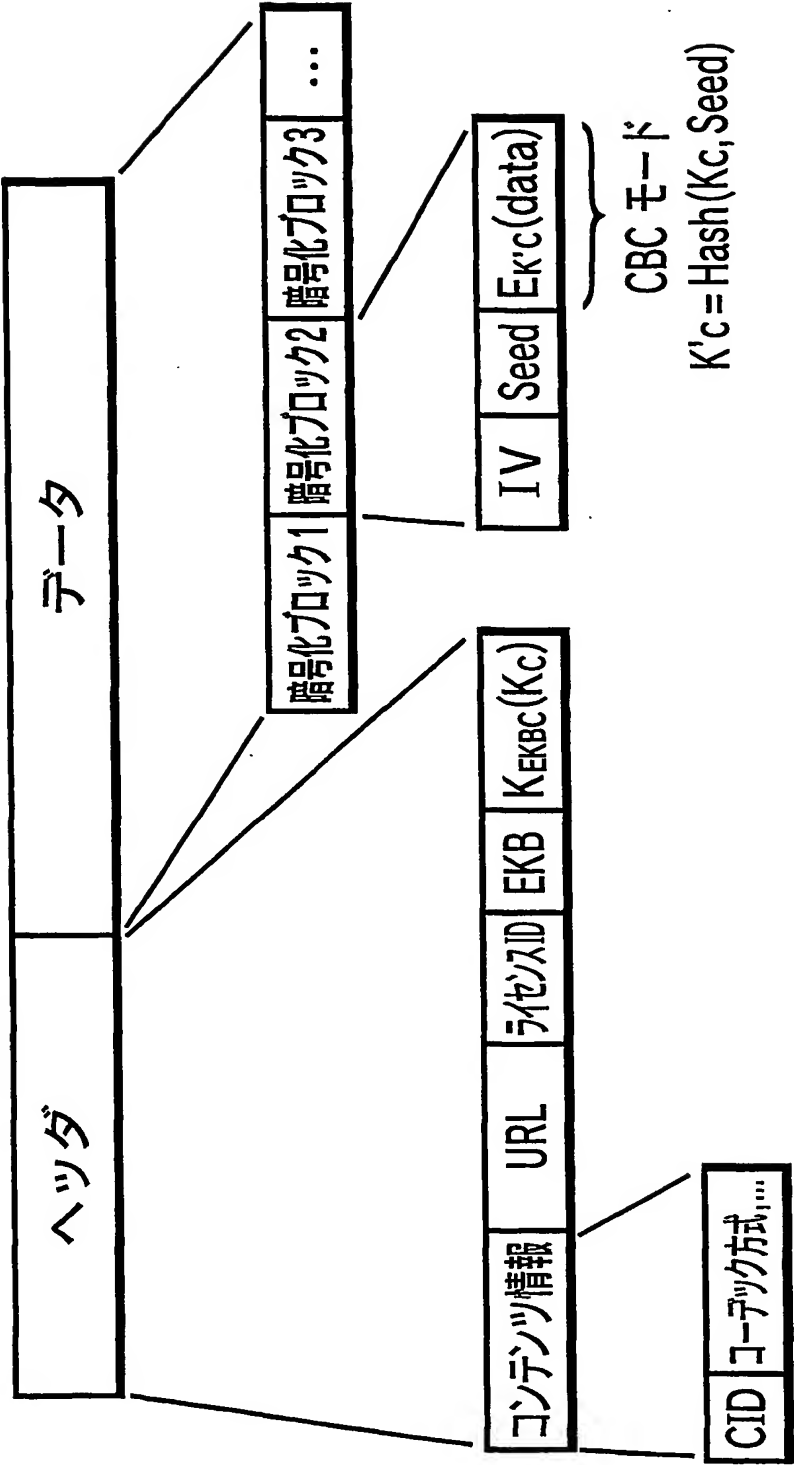
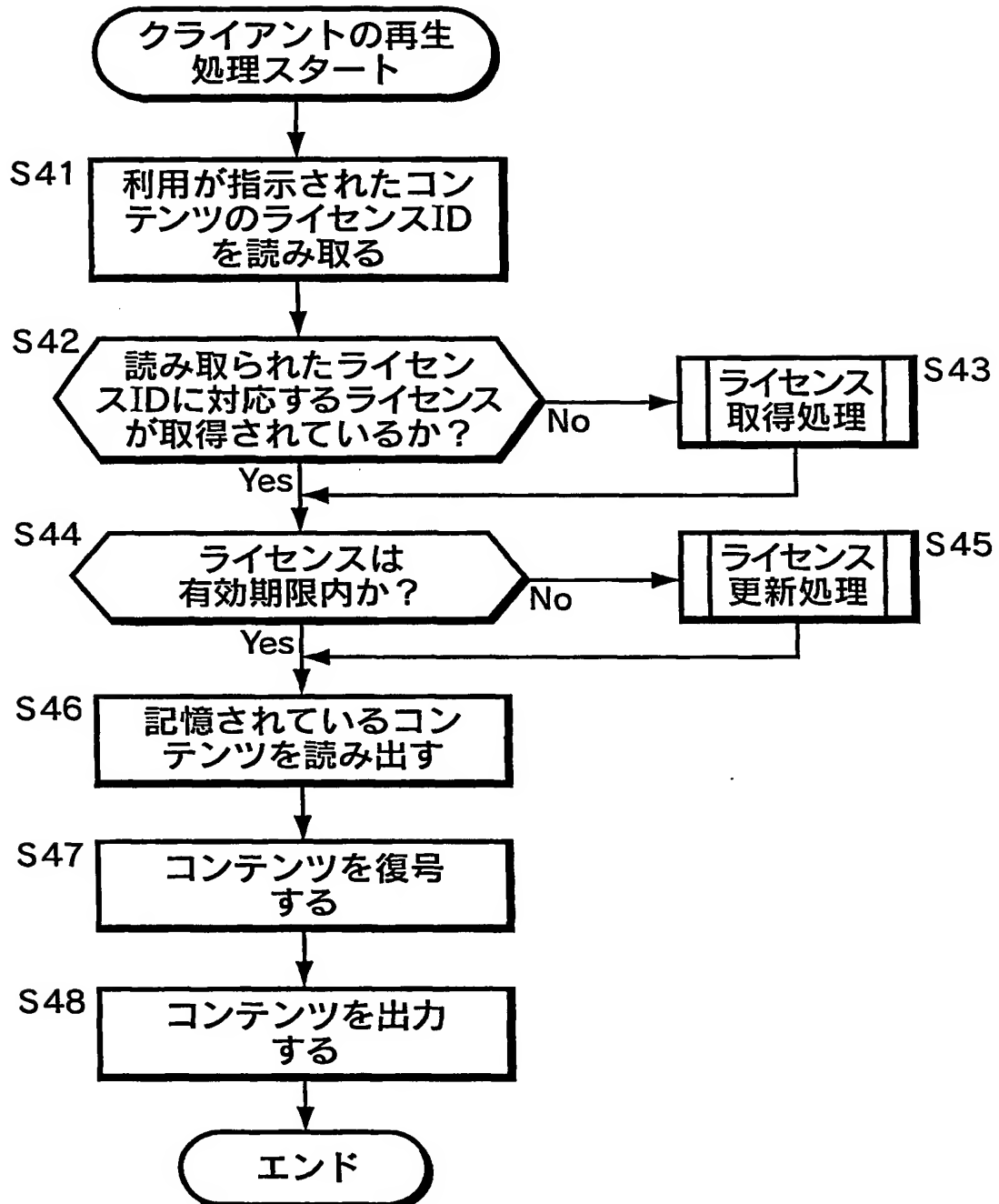


図5



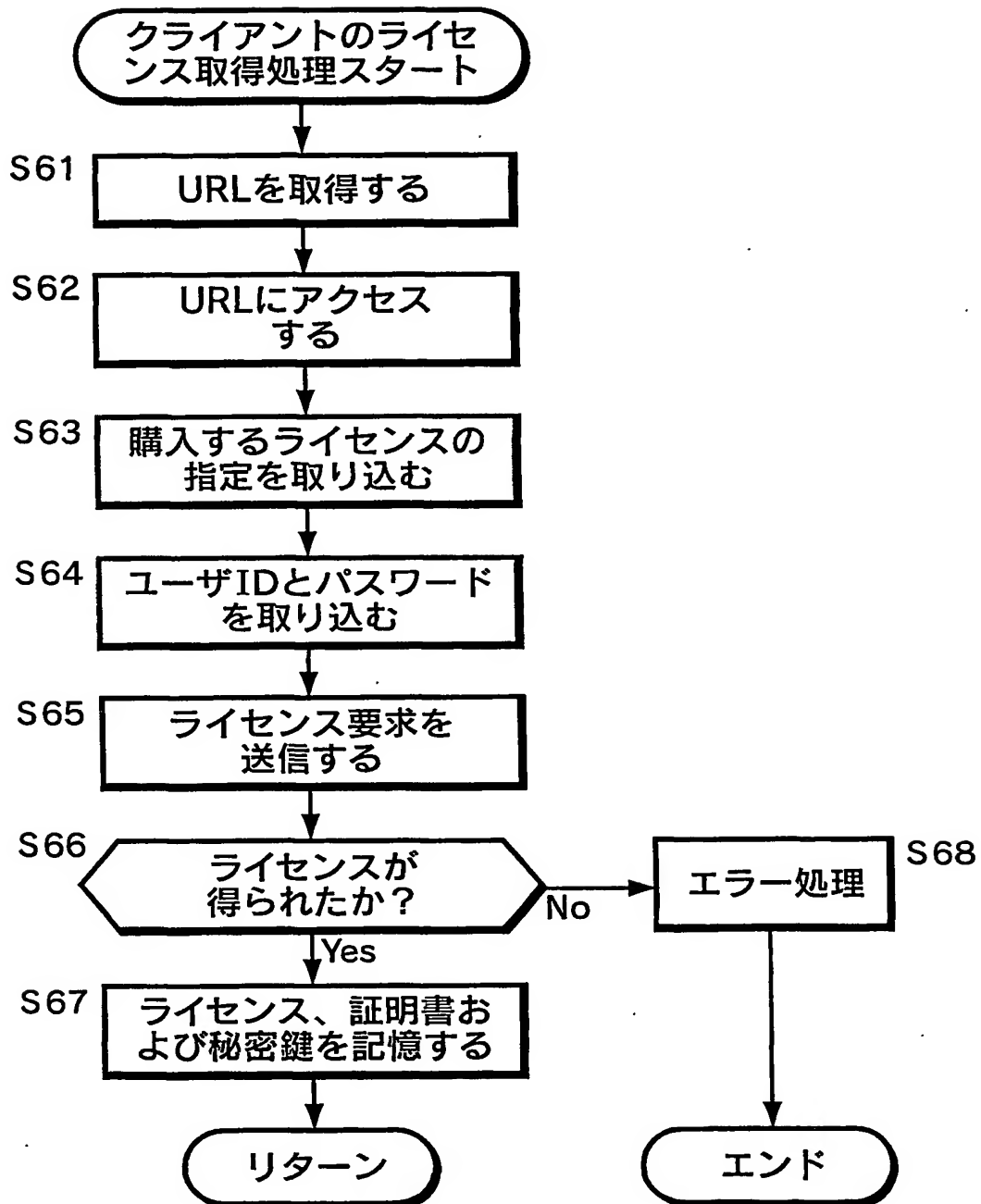
6/45

図 6



7/45

図 7



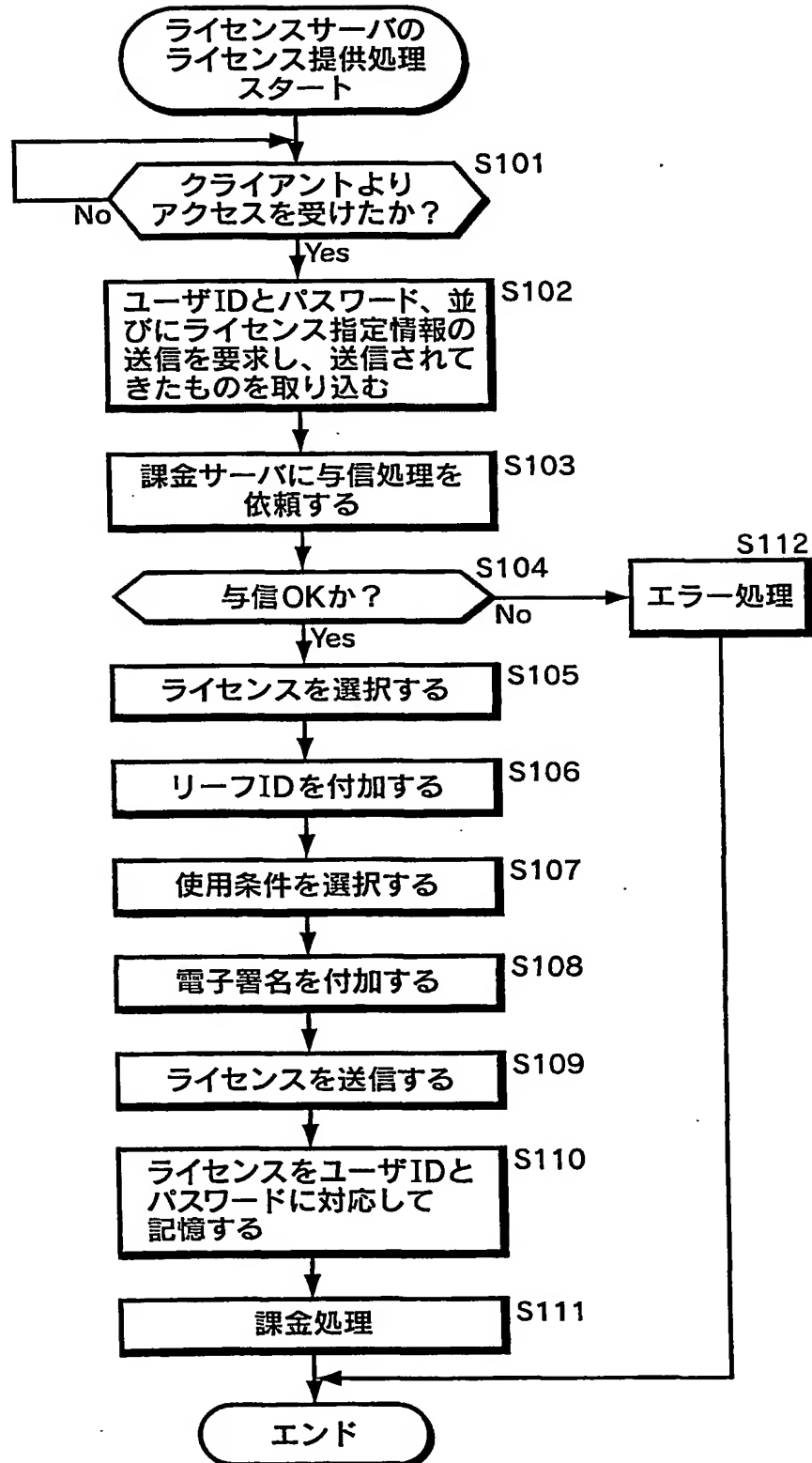
8/45

図 8

ライセンスID
作成日時
有効期限
使用条件
リーフID
電子署名
ライセンス

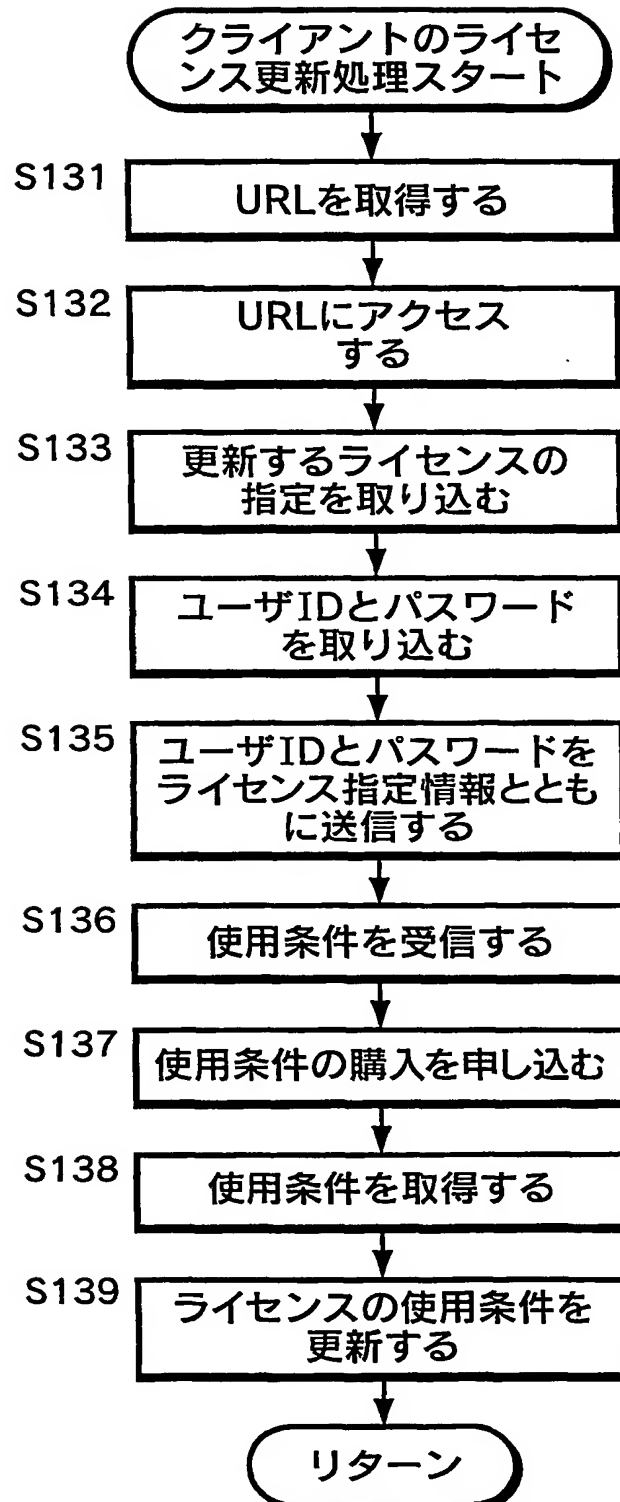
9/45

図 9



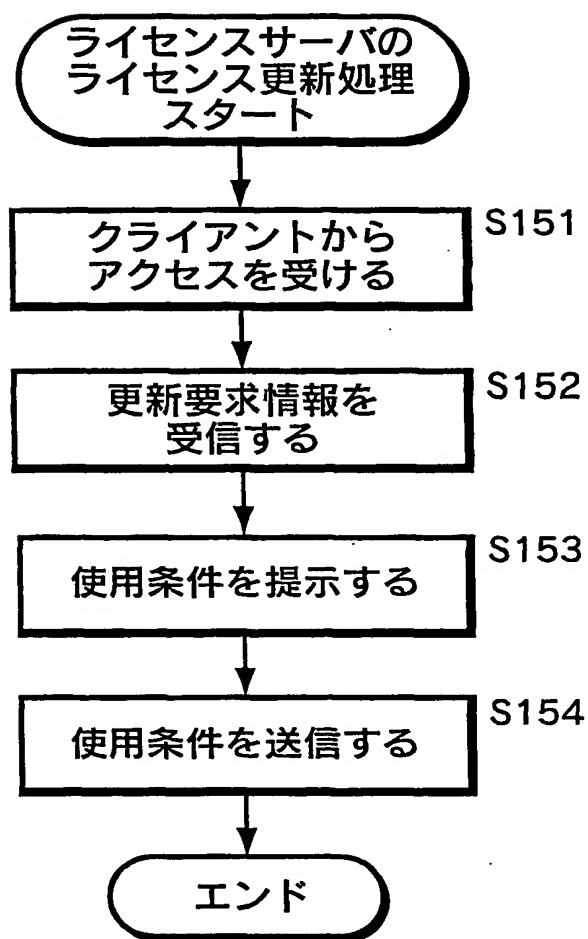
10/45

図 10



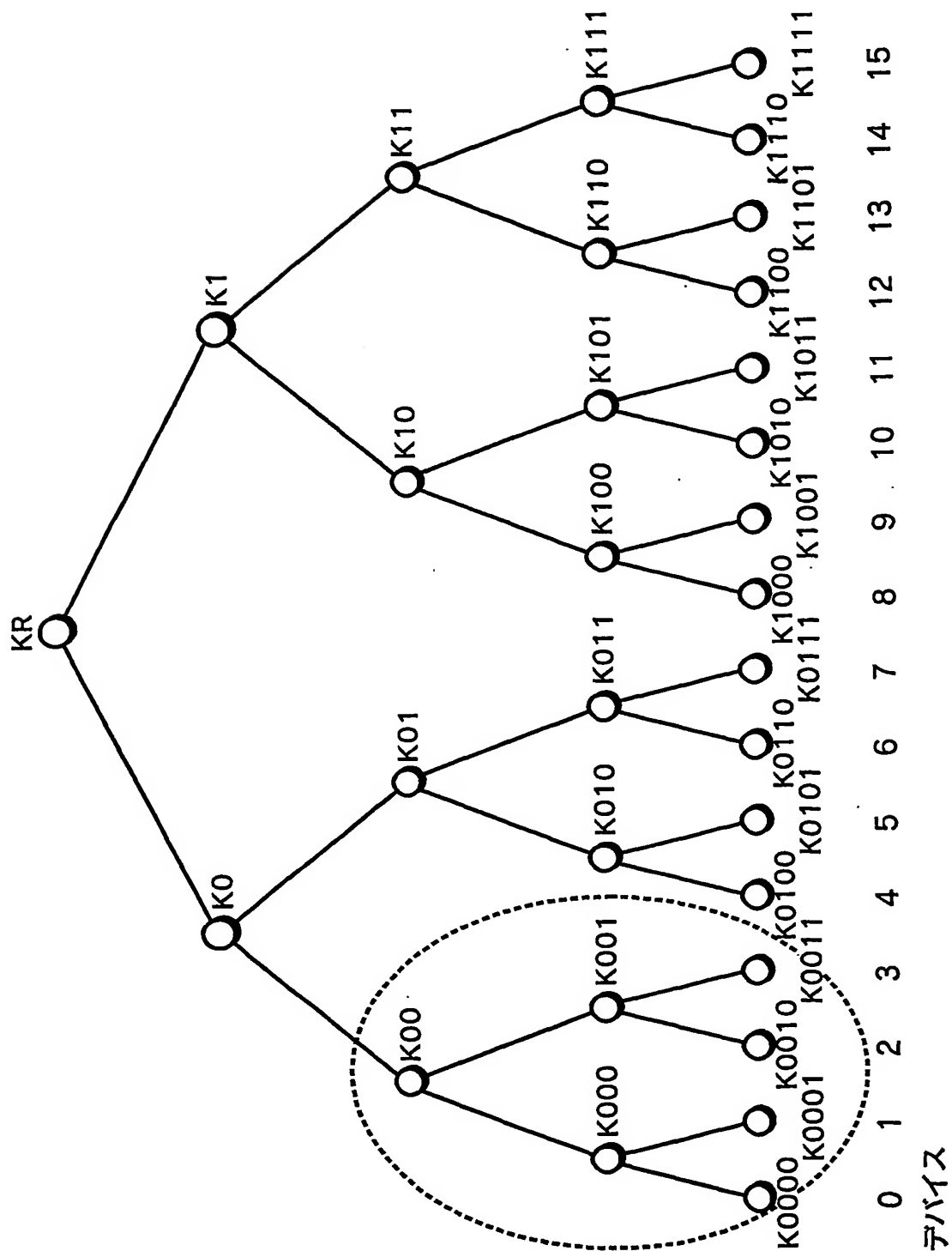
11/45

図 11



12/45

図12



13/45

図 13

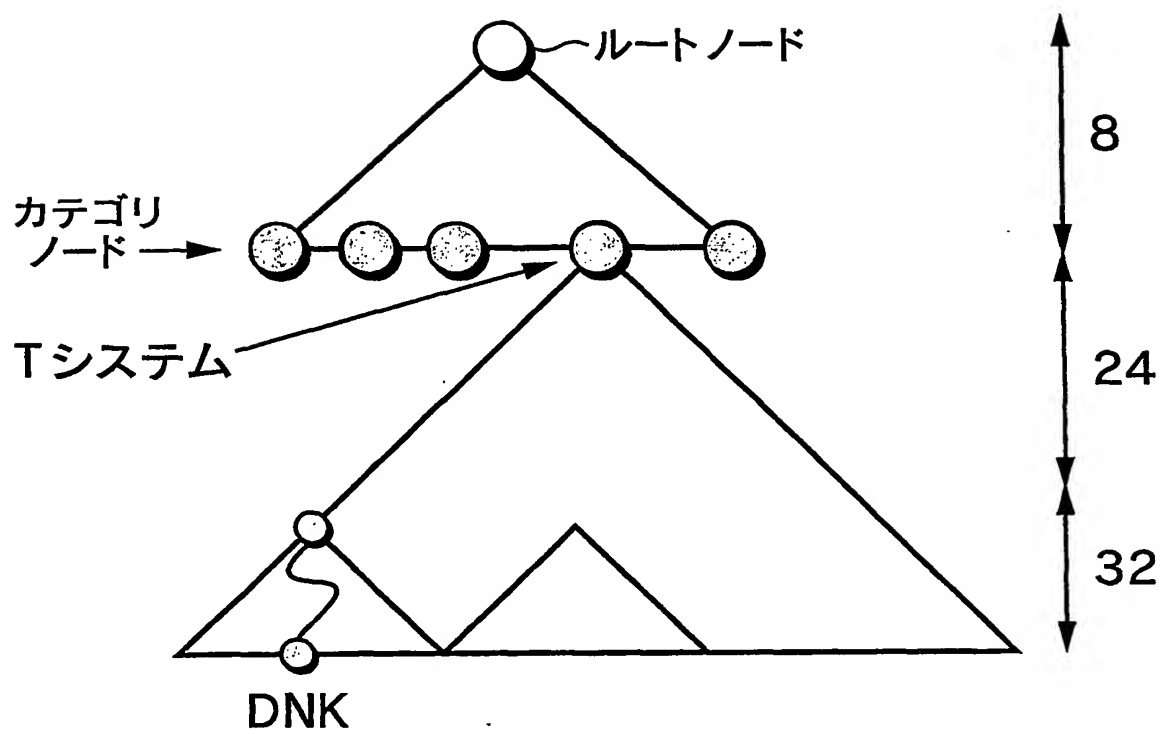
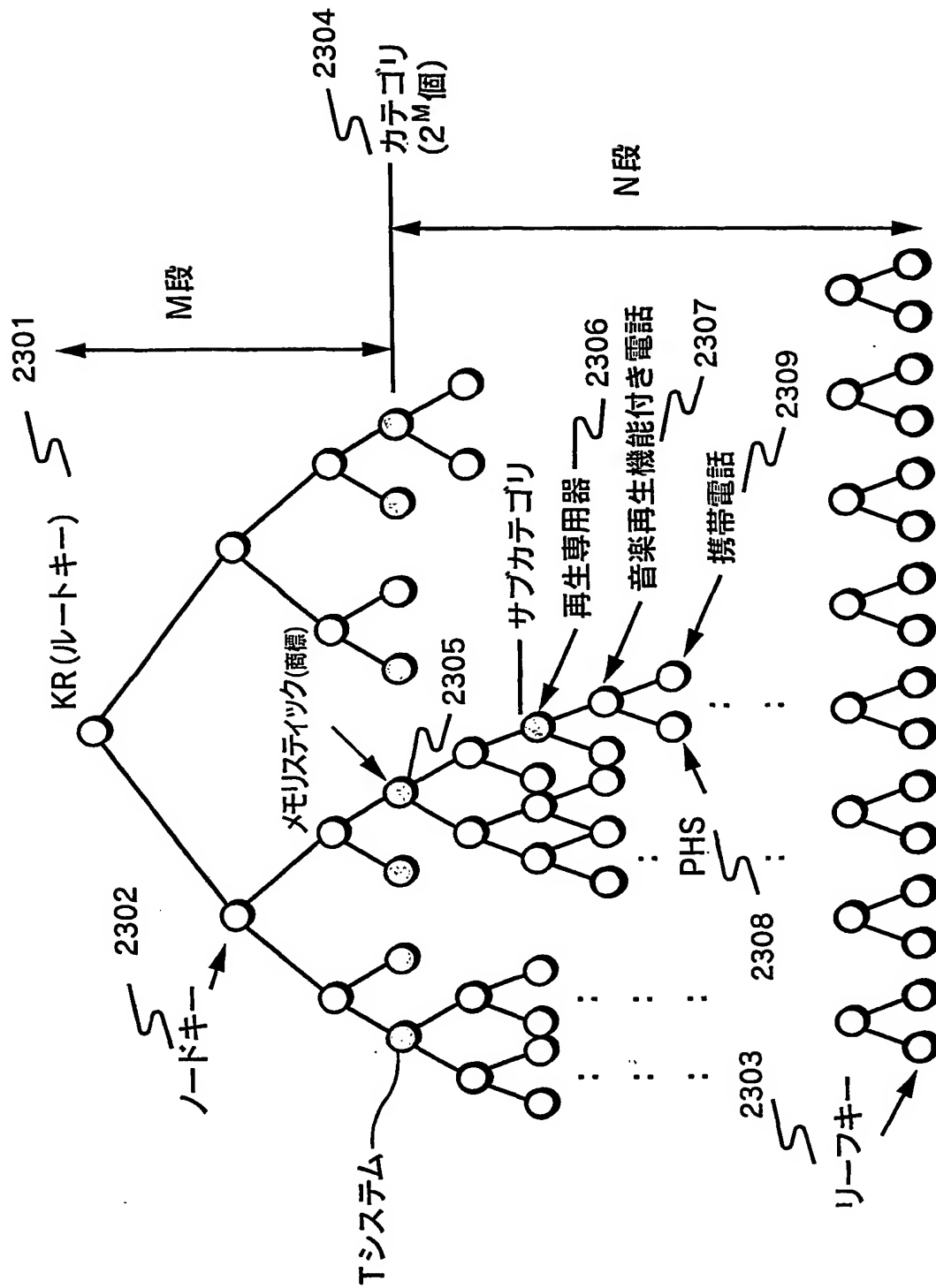


図14



15/45

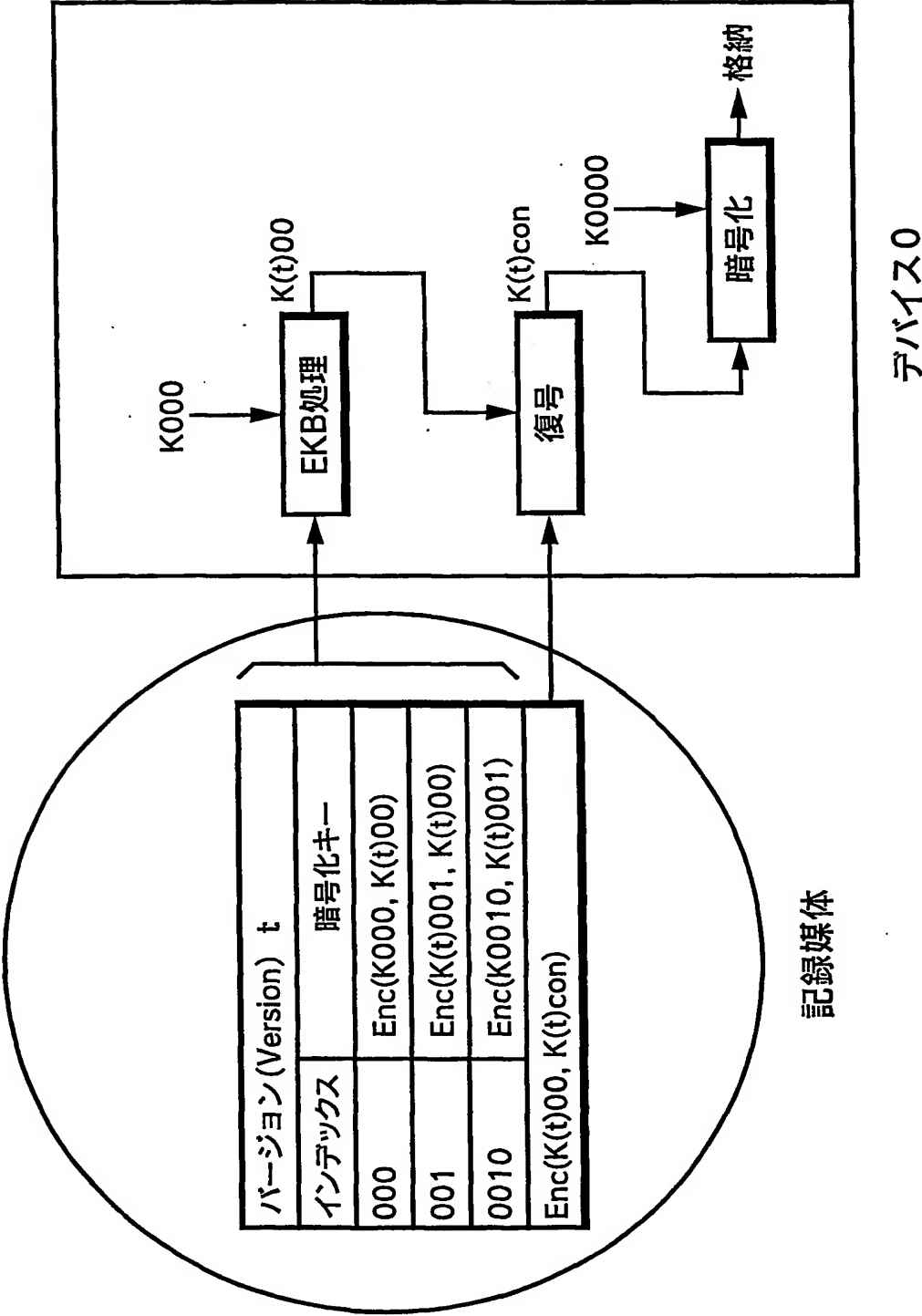
図15A

バージョン (Version) t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

図15B

バージョン (Version) t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

図16



17/45

図 17

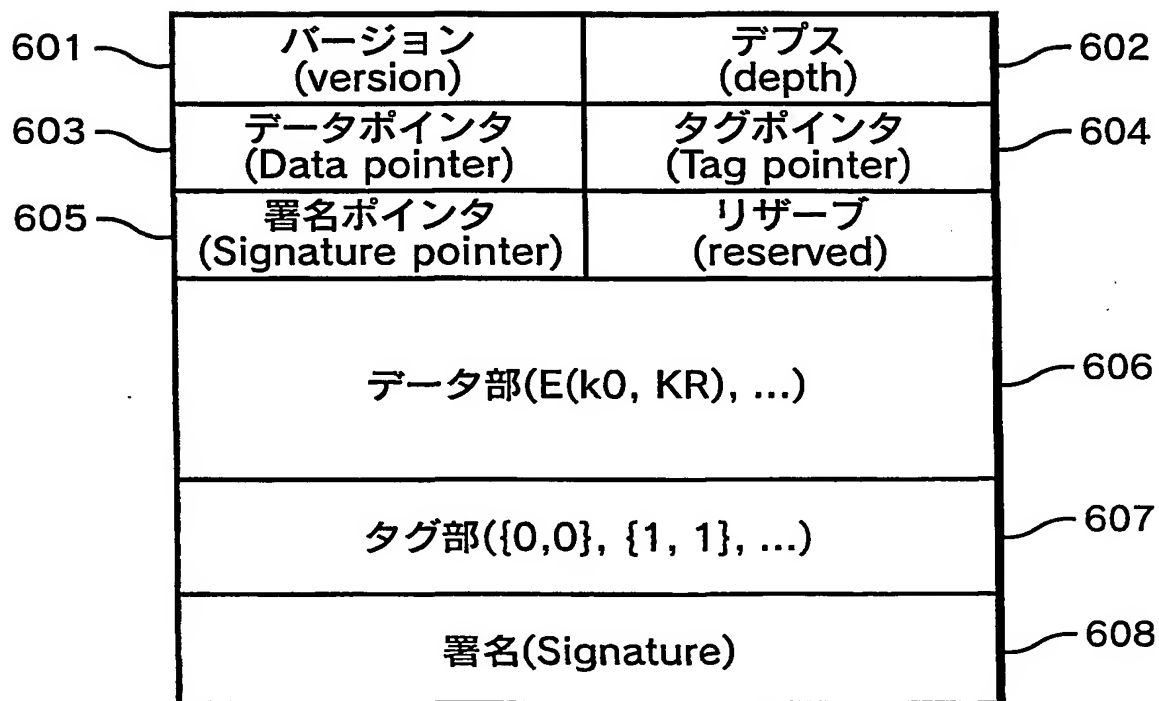
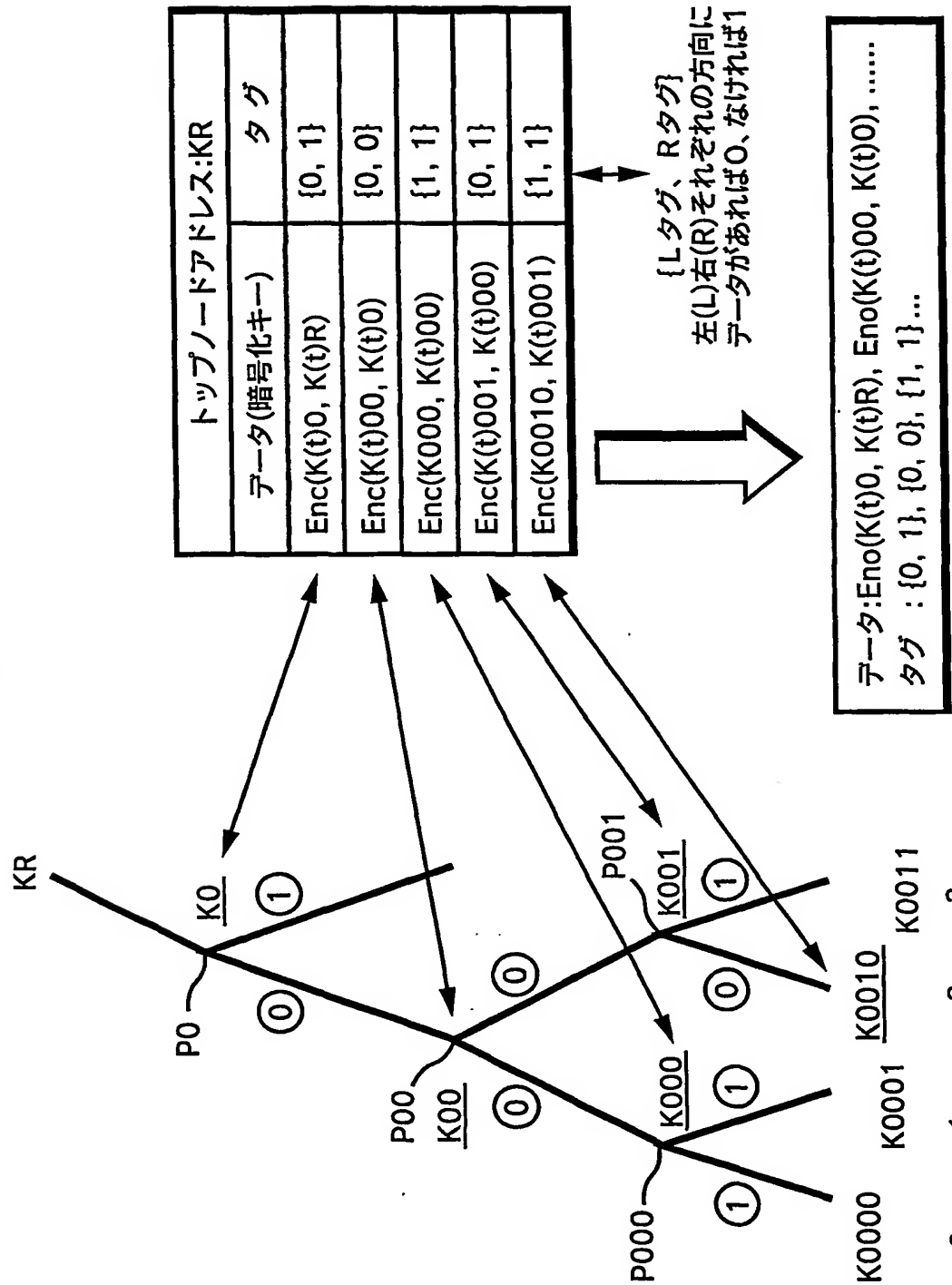


図18



19/45

図 19

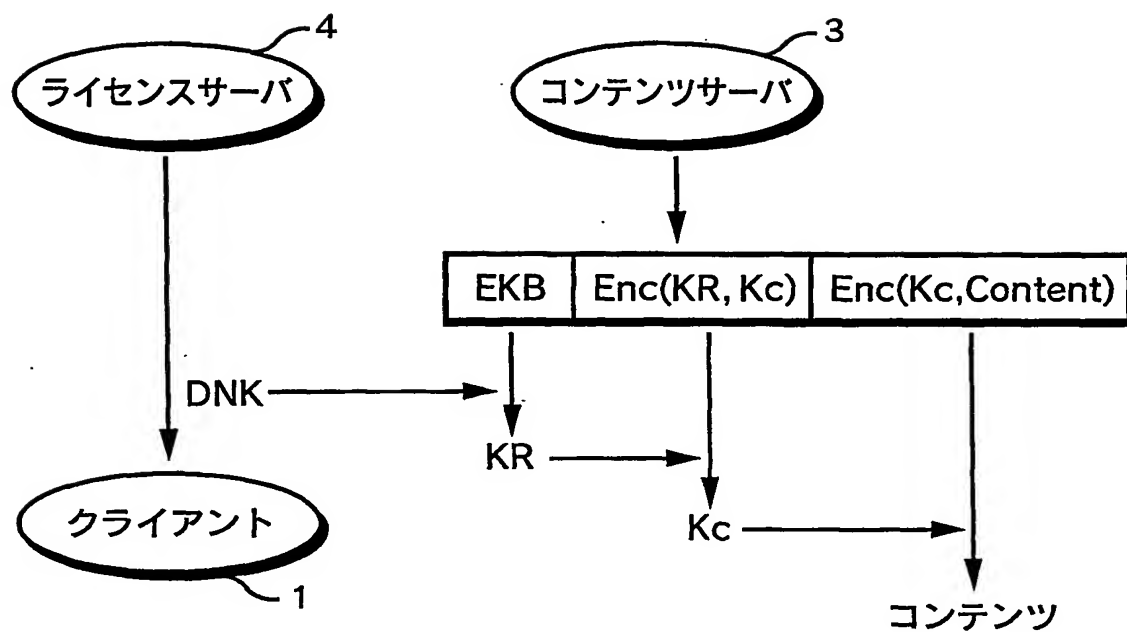


図 20

EKB



20/45

図 21

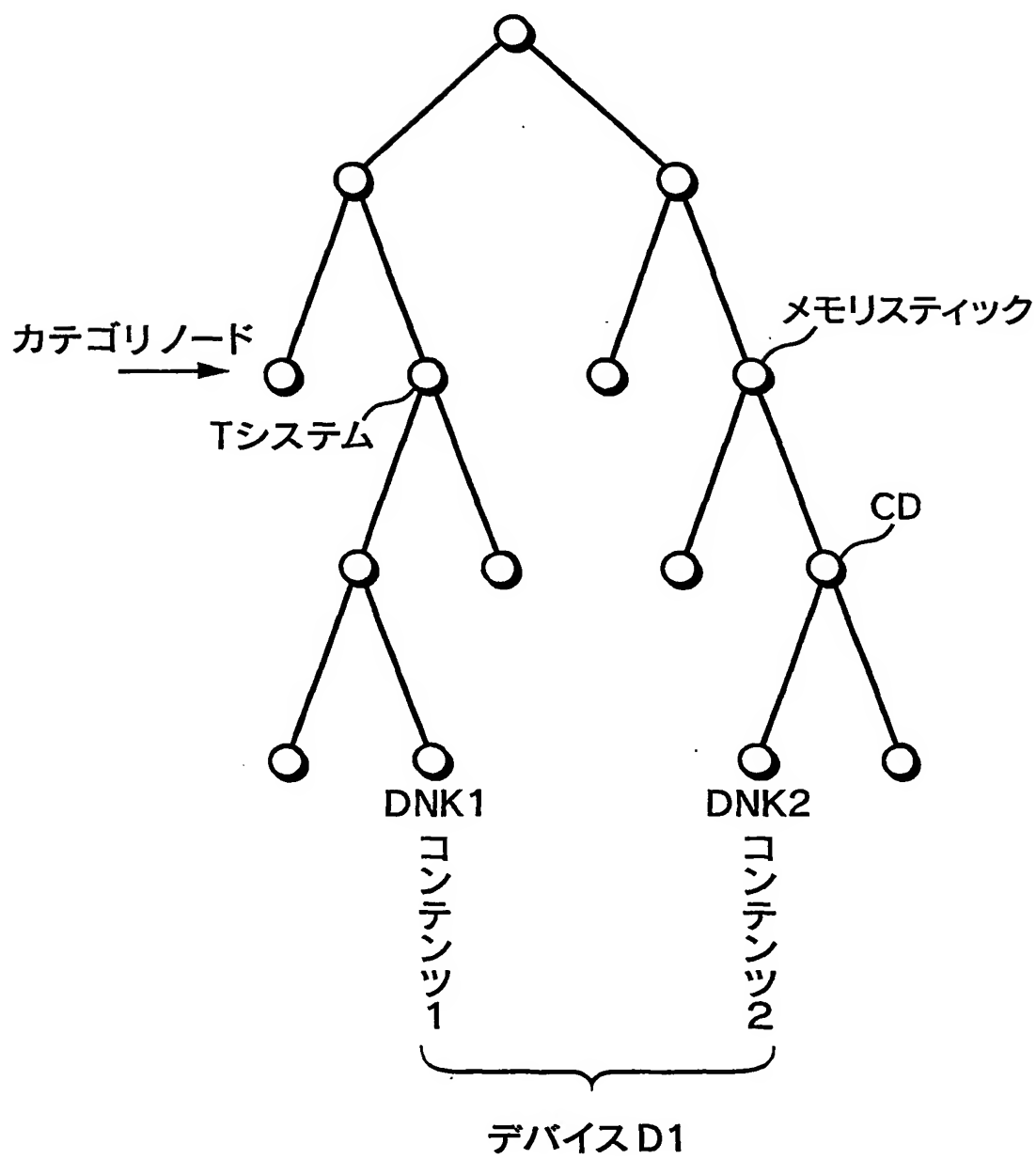
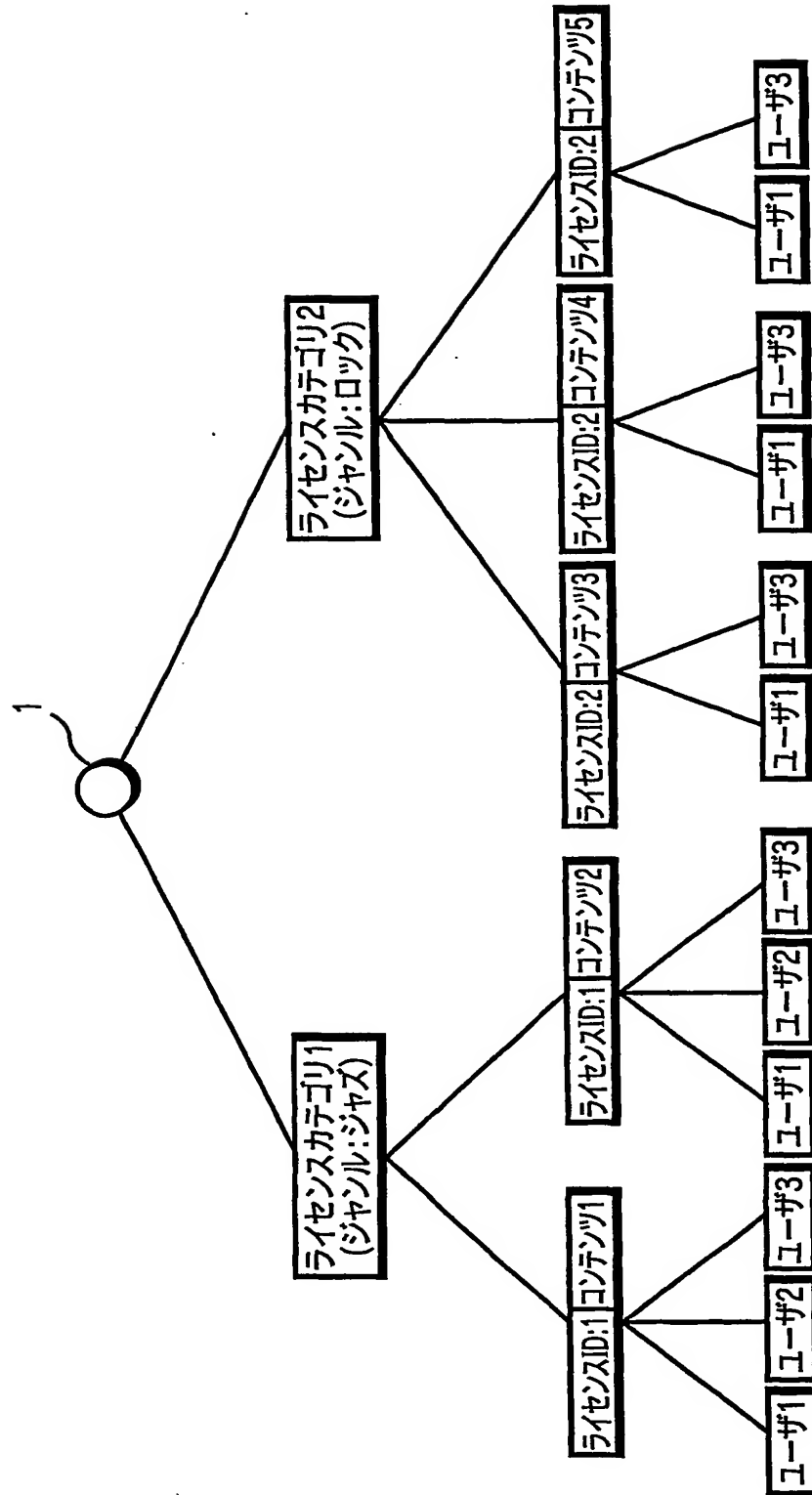
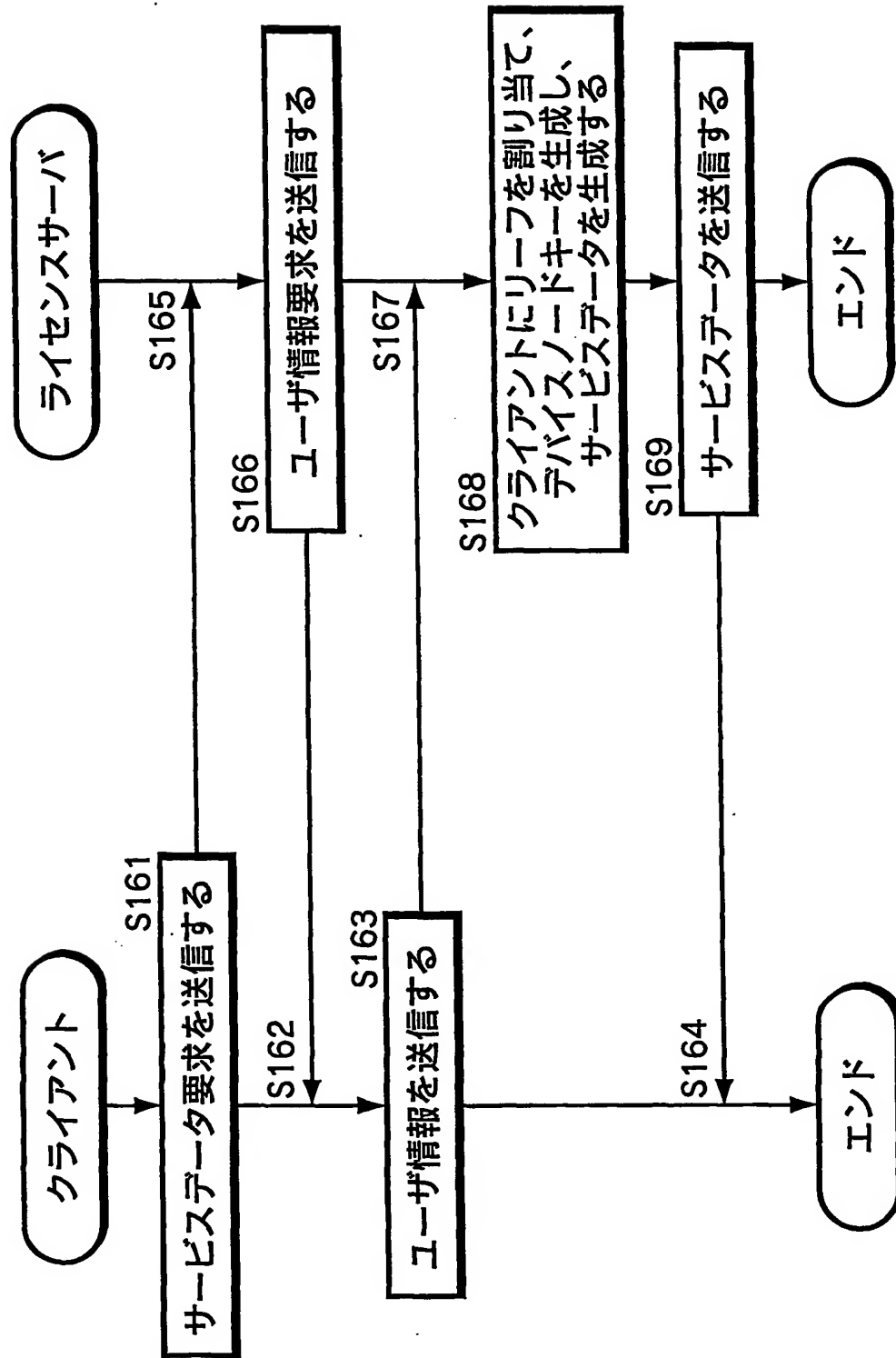


図 22



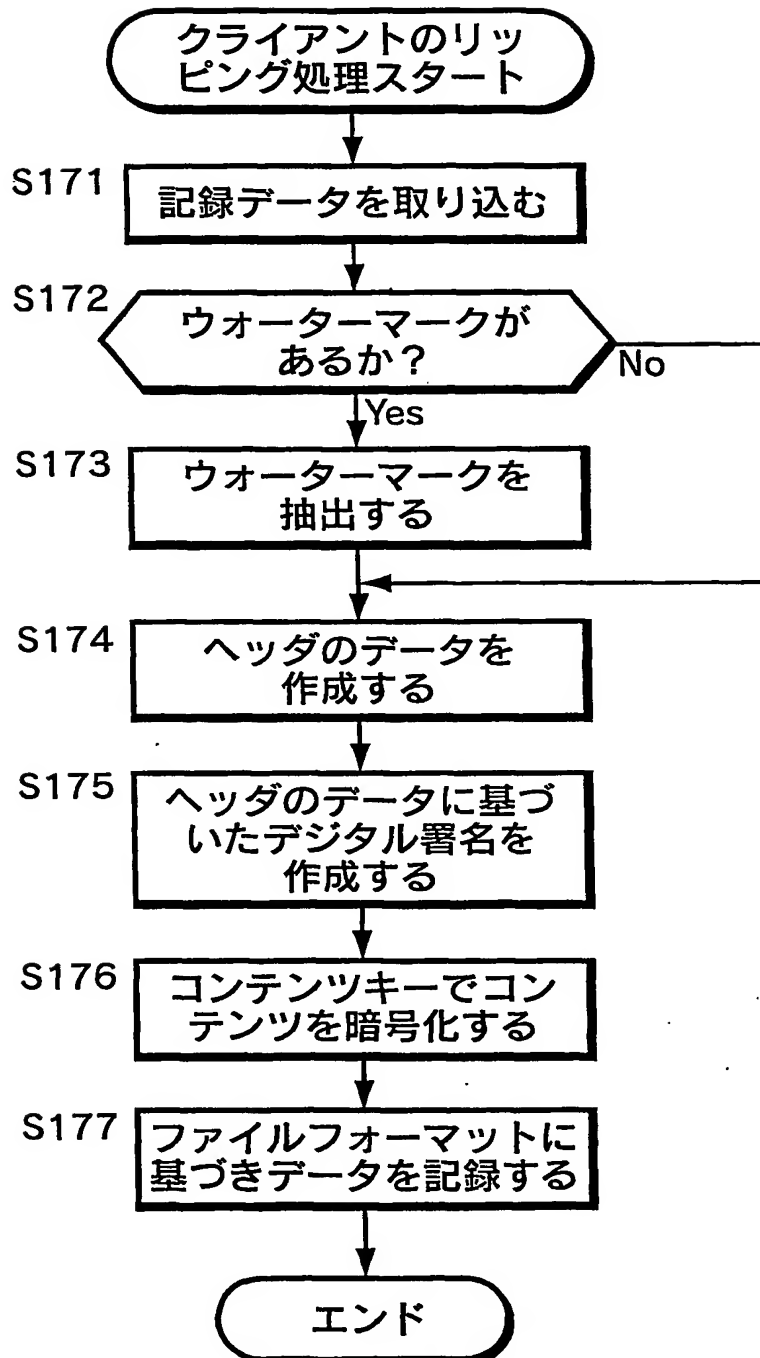
22/45

図 23



23/45

図 24



24/45

図 25

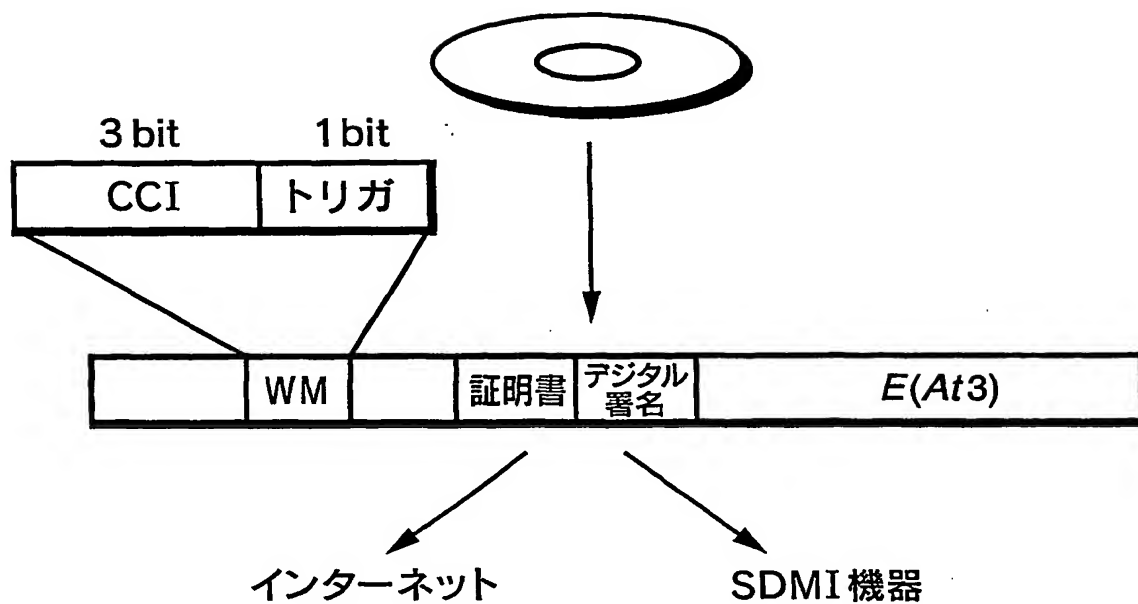
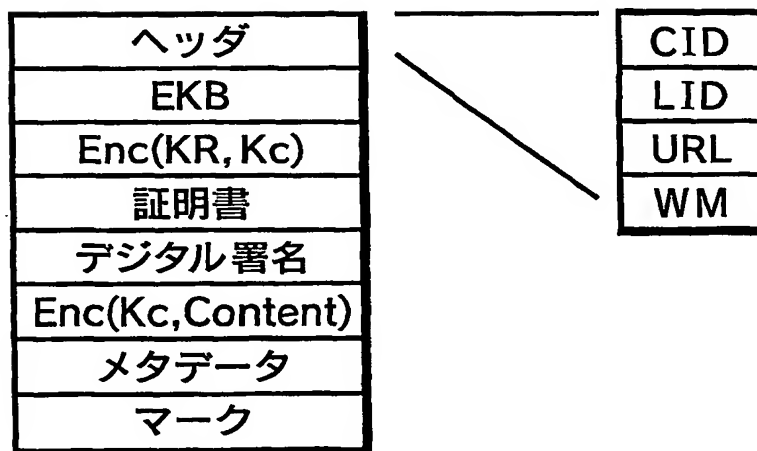
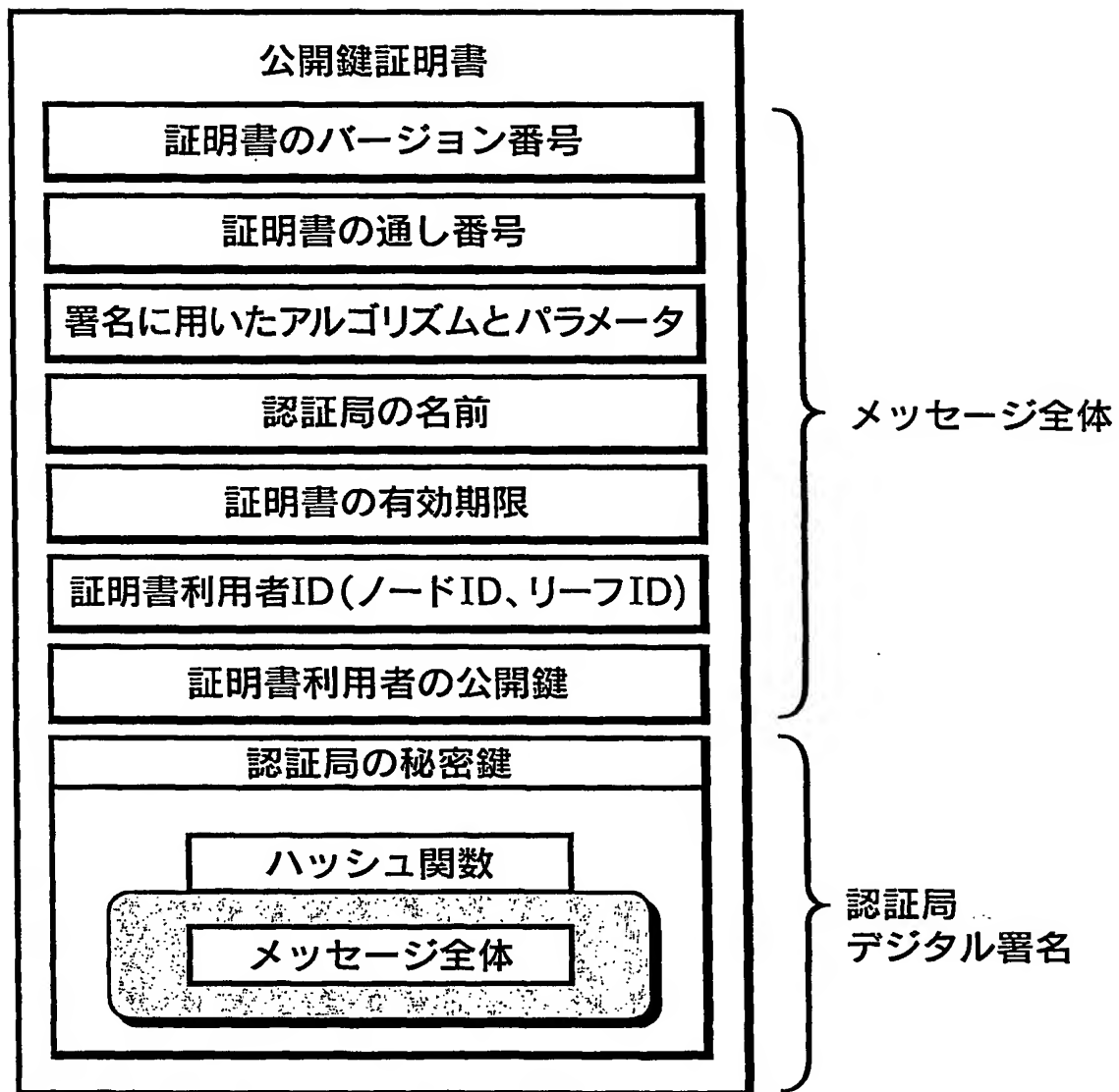


図 26



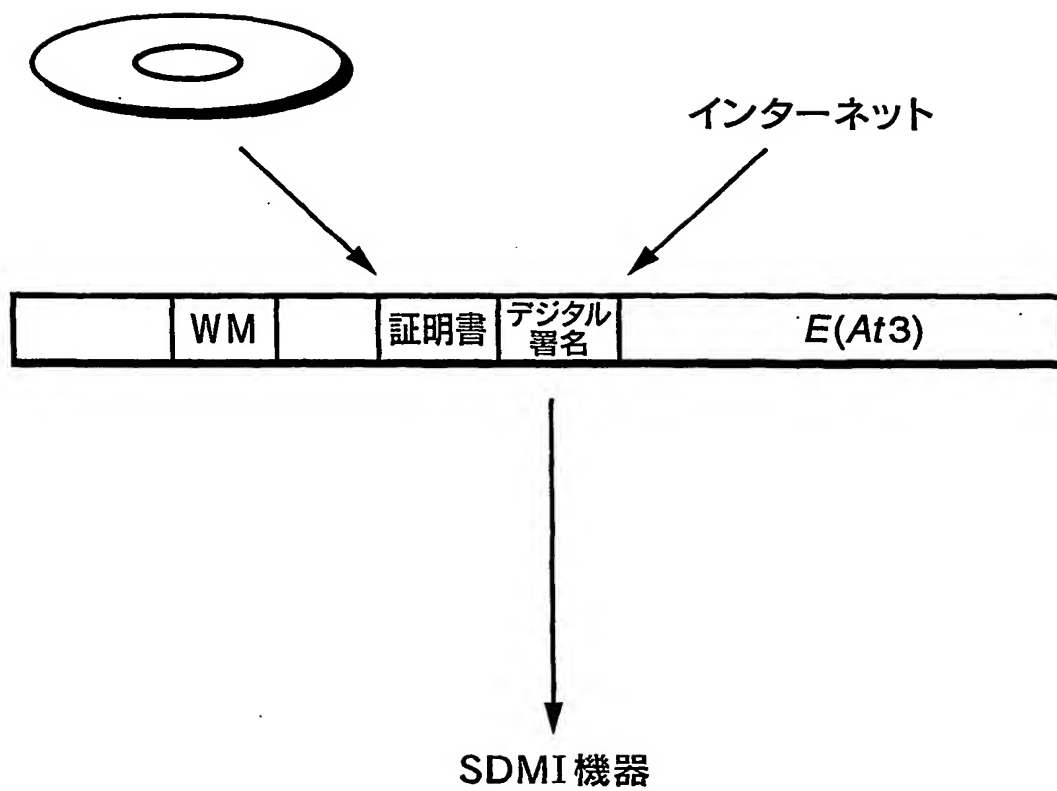
25/45

図 27



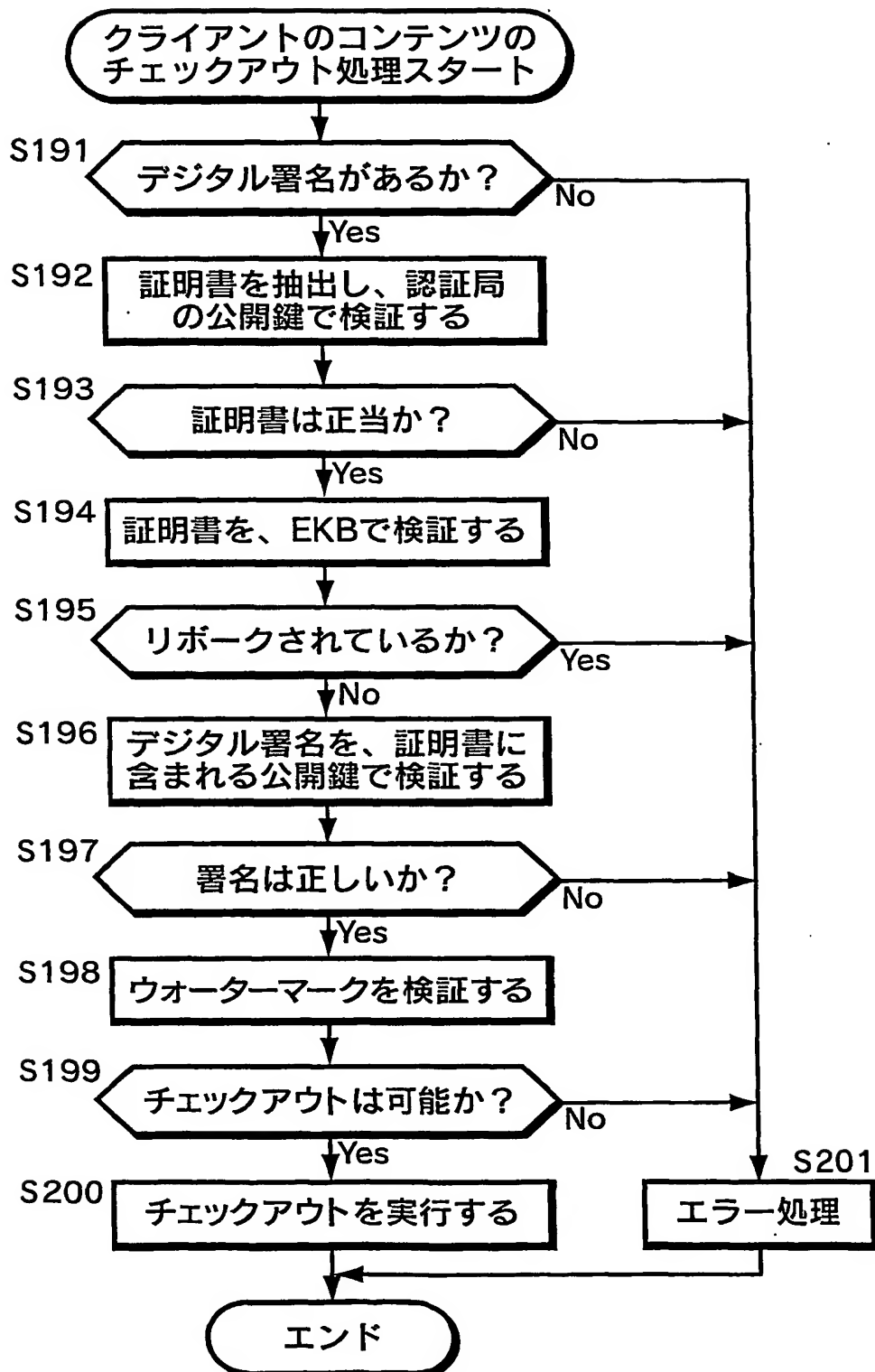
26/45

図 28



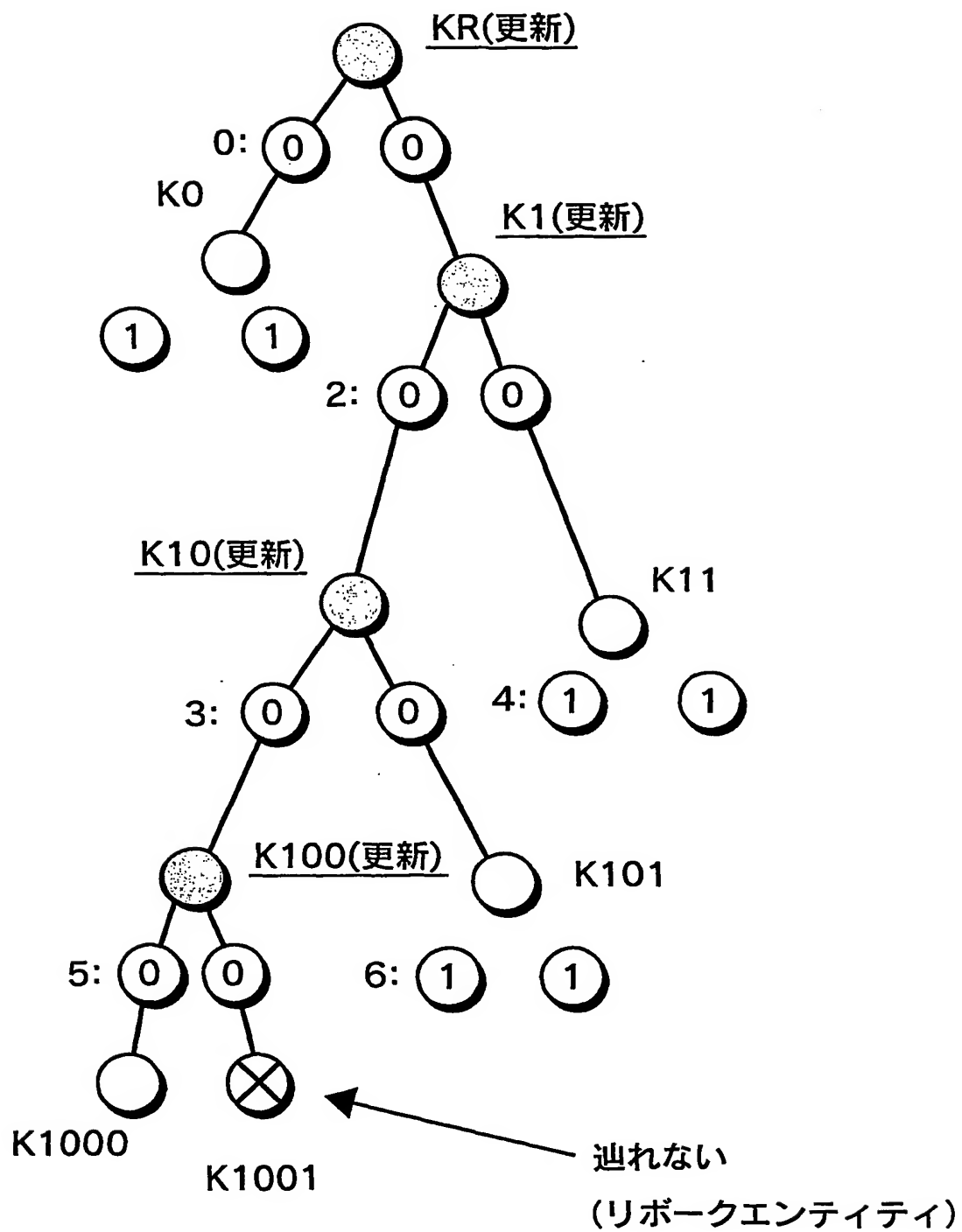
27/45

図 29



28/45

図 30



29/45

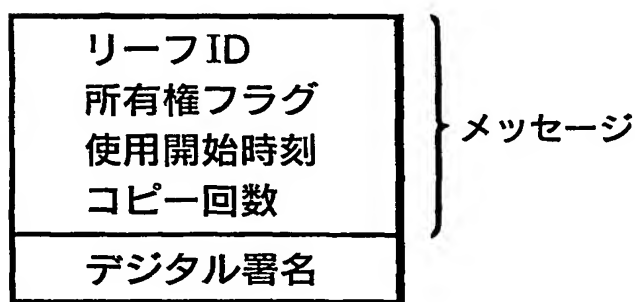
図 31

データ (暗号化キー)	$\text{Enc}(K0, K(t)R), \text{Enc}(K(t)1, K(t)R)$ $\text{Enc}(K(t)10, K(t)1), \text{Enc}(K11, K(t)1)$ $\text{Enc}(K(t)100, K(t)10), \text{Enc}(K101, K(t)10)$ $\text{Enc}(K1000, K(t)100)$
タグ	0: {0, 0}, 1:{1, 1}, 2:{0, 0}, 3:{0, 0} 4: {1, 1}, 5:{0, 1}, 6:{1, 1}



{L タグ、R タグ}
 左(L)右(R)それぞれの方向に
 データがあれば0、なければ1

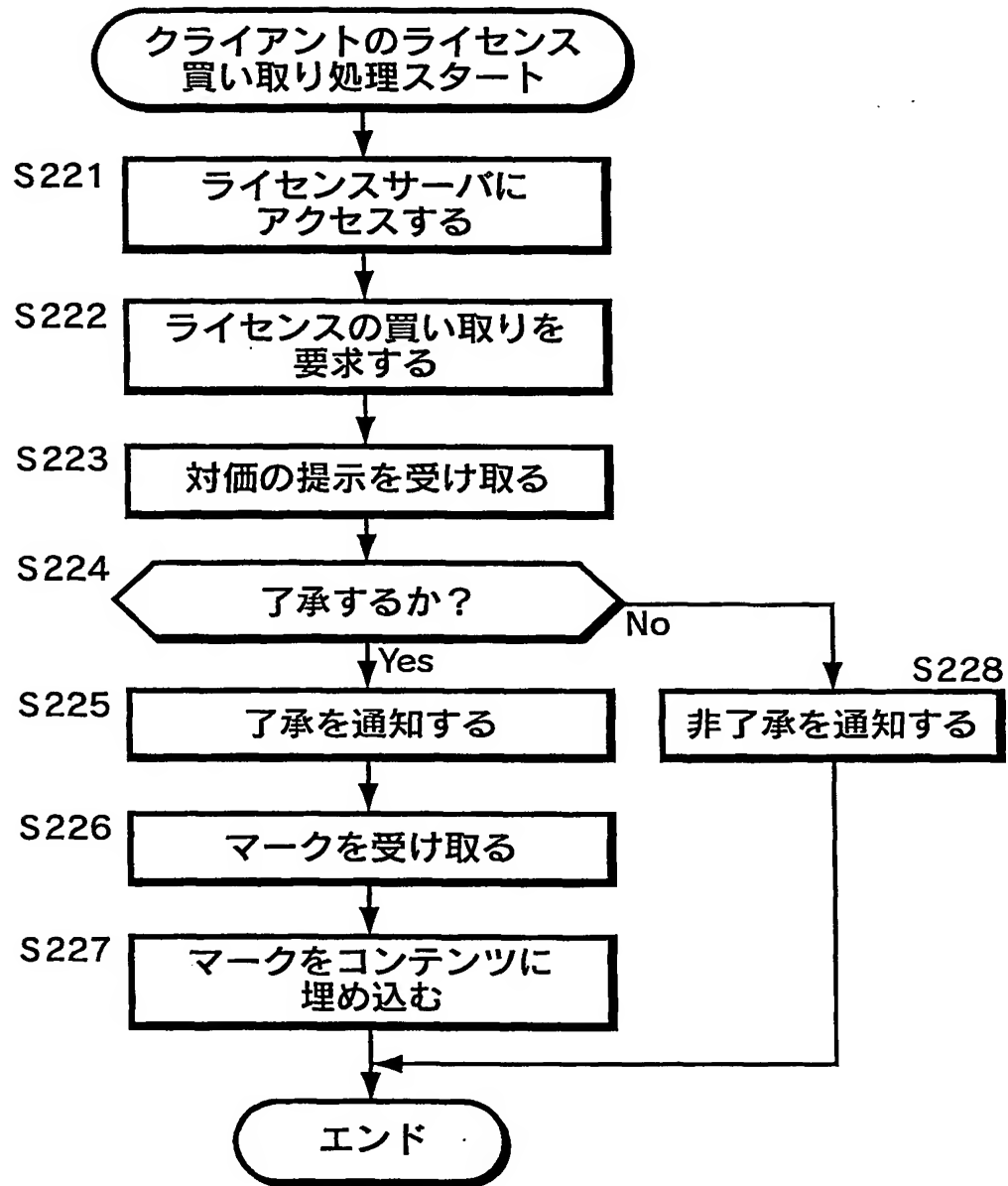
図 32



マーク

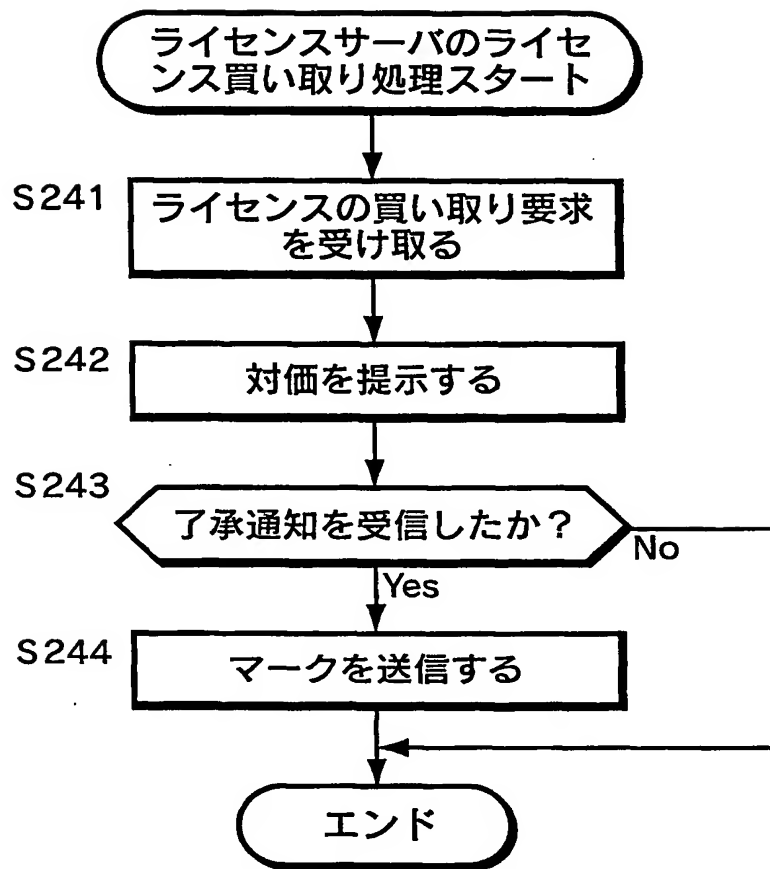
30/45

図 33



31/45

図 34

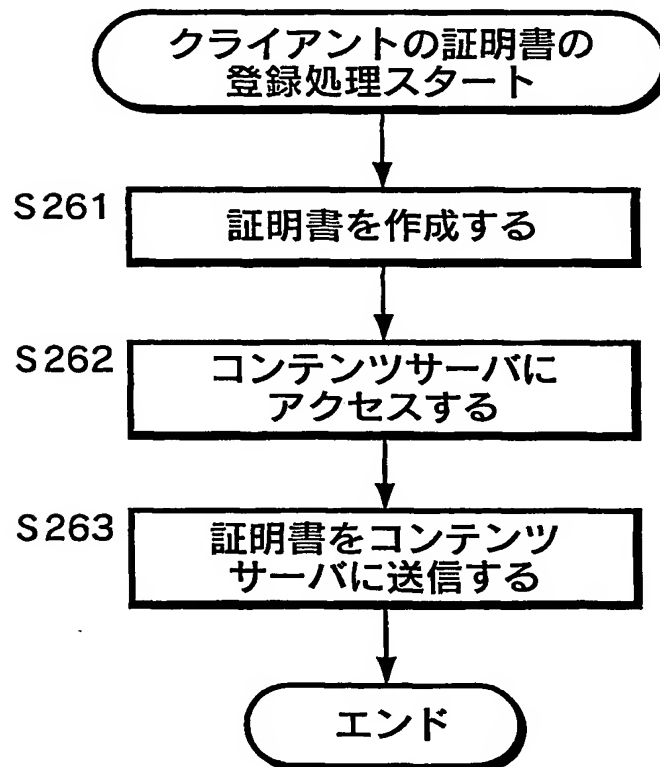


32/45

図 35

$$\text{Mark} = \{\text{LeafID}, \text{Own}, \text{Sigs}(\text{LeafID}, \text{Own})\}$$

図 36



33/45

図 37

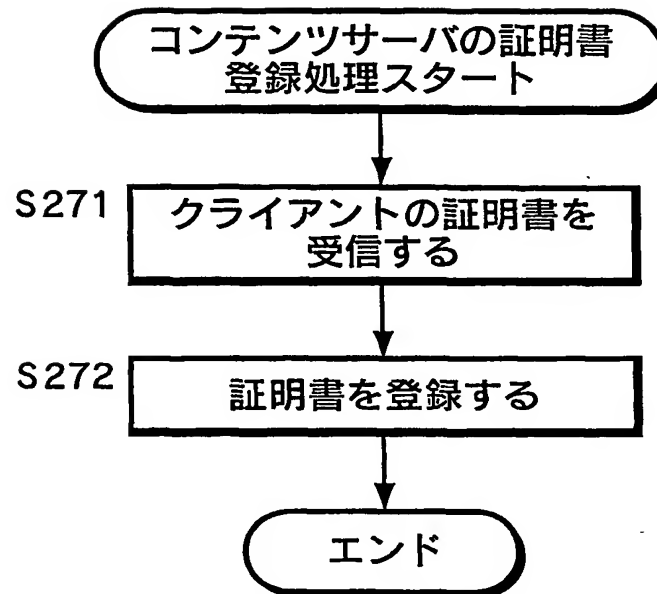
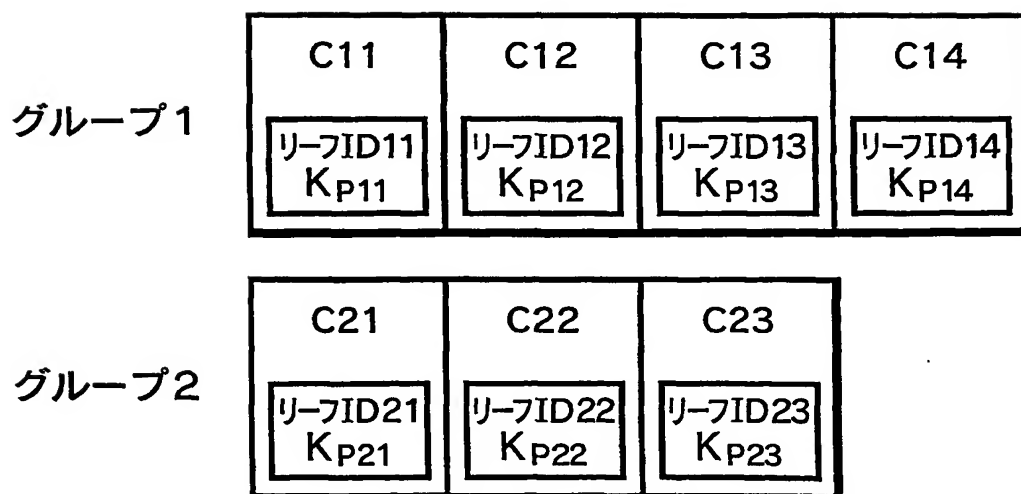


図 38



34/45

図 39

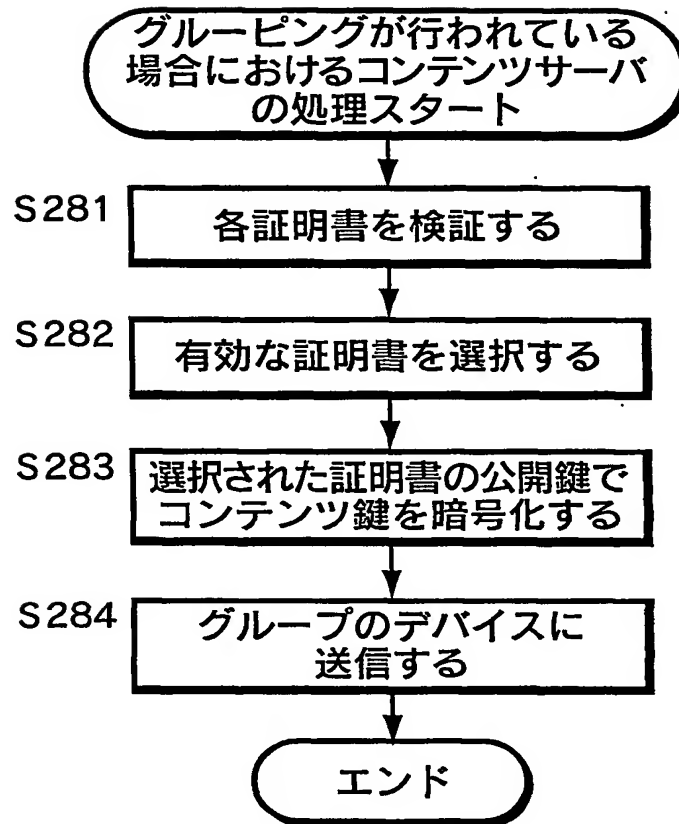
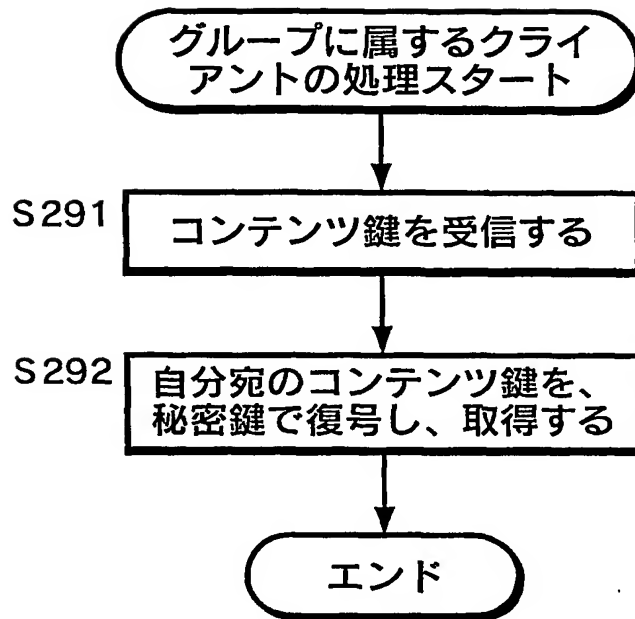


図 40

$$\text{Enc}(K_{P11}, K_C), \text{Enc}(K_{P12}, K_C), \text{Enc}(K_{P13}, K_C)$$

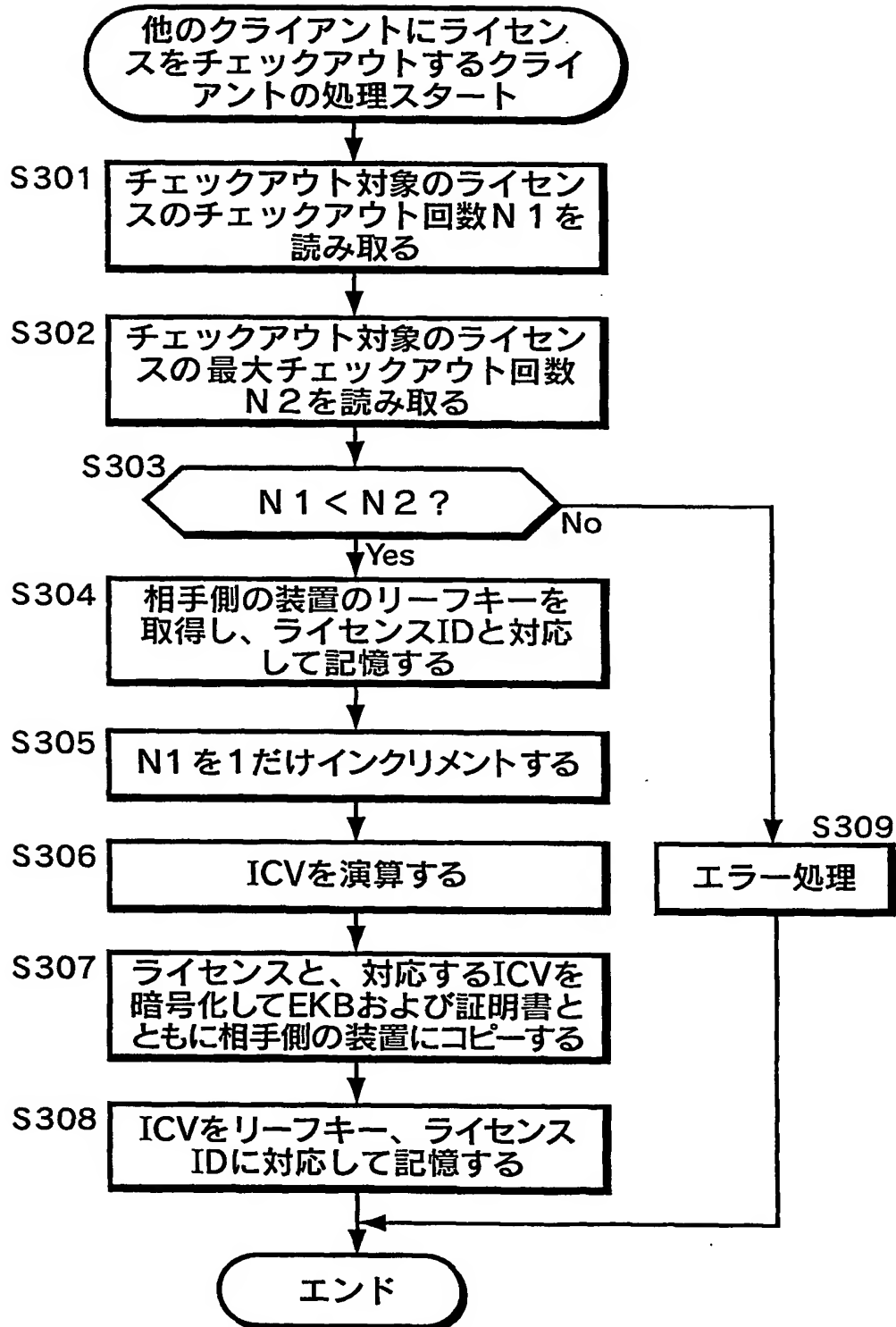
35/45

図 41



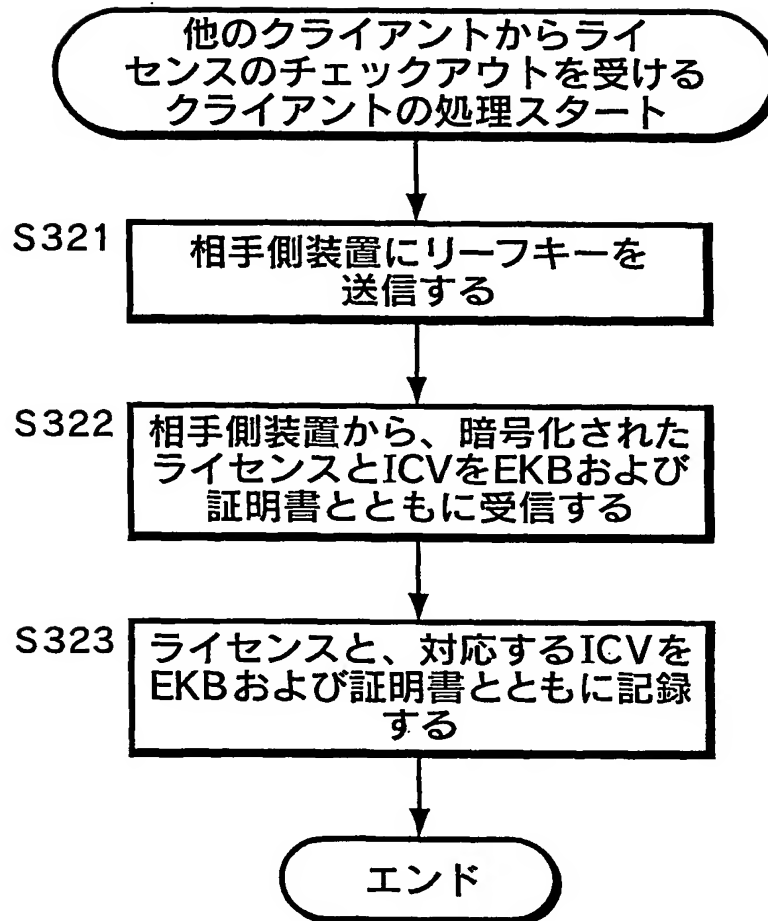
36/45

図 42



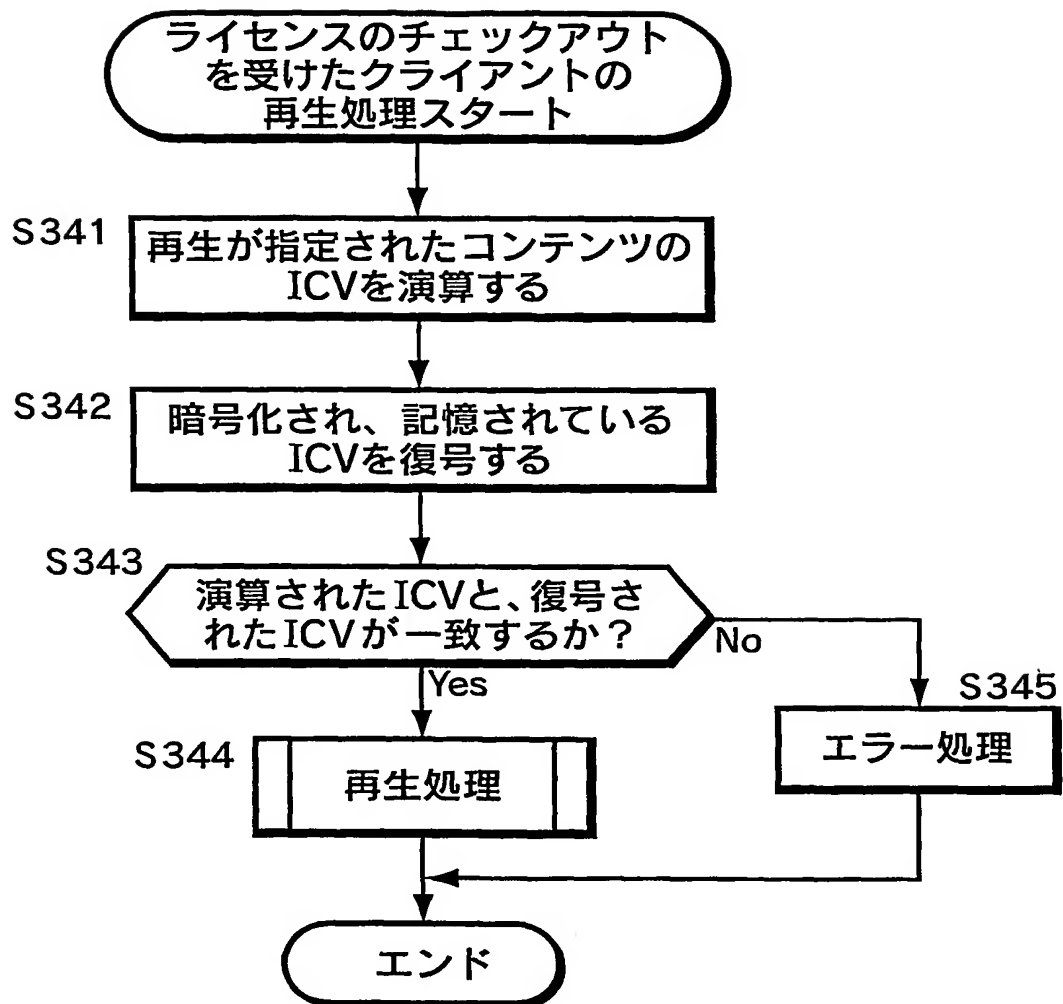
37/45

図 43



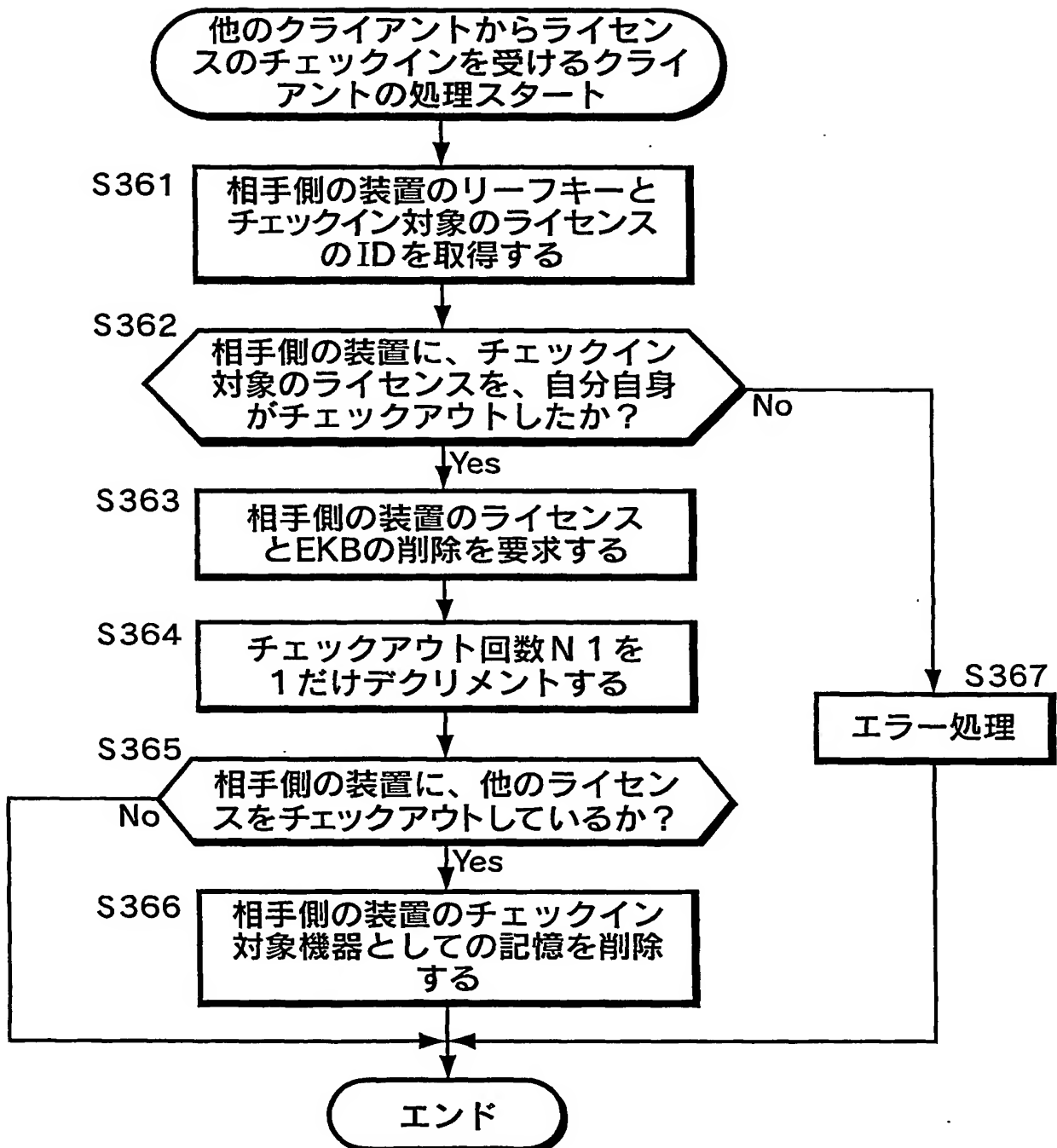
38/45

図 44



39/45

図 45



40/45

図 46

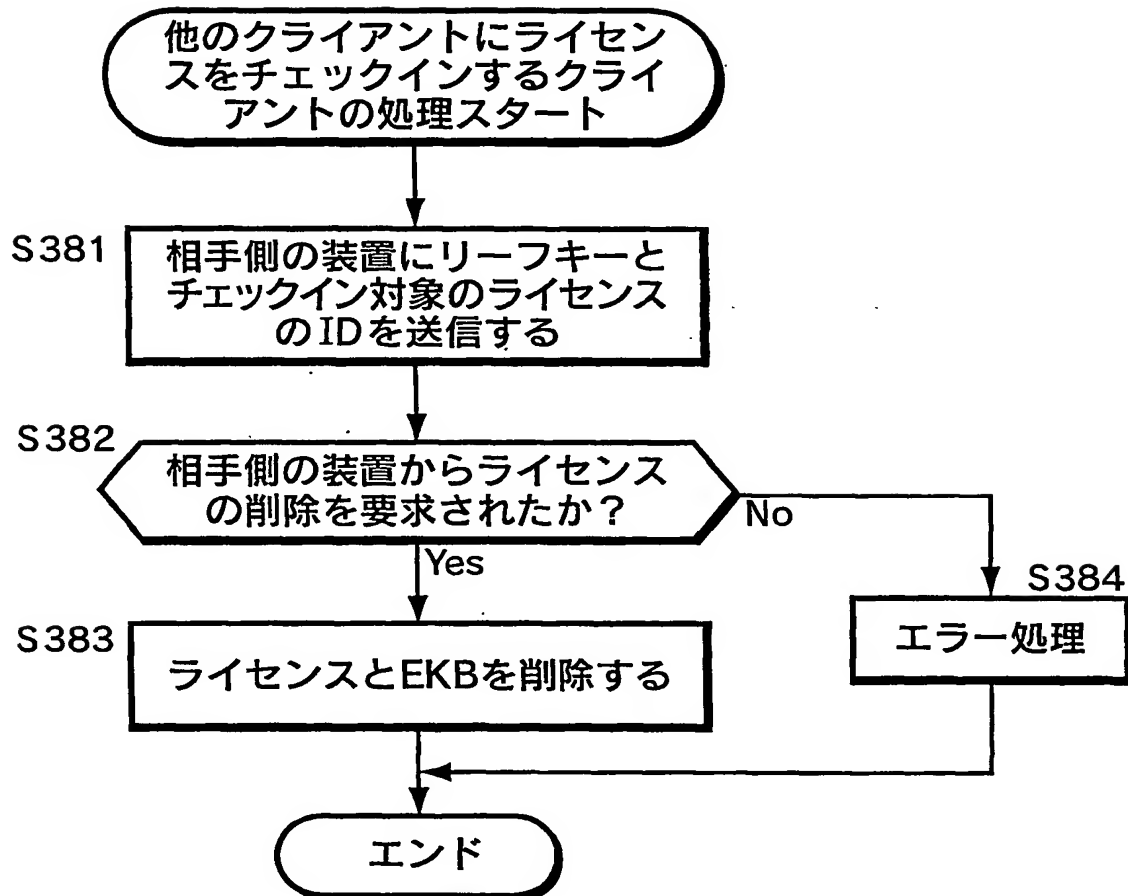


図 47

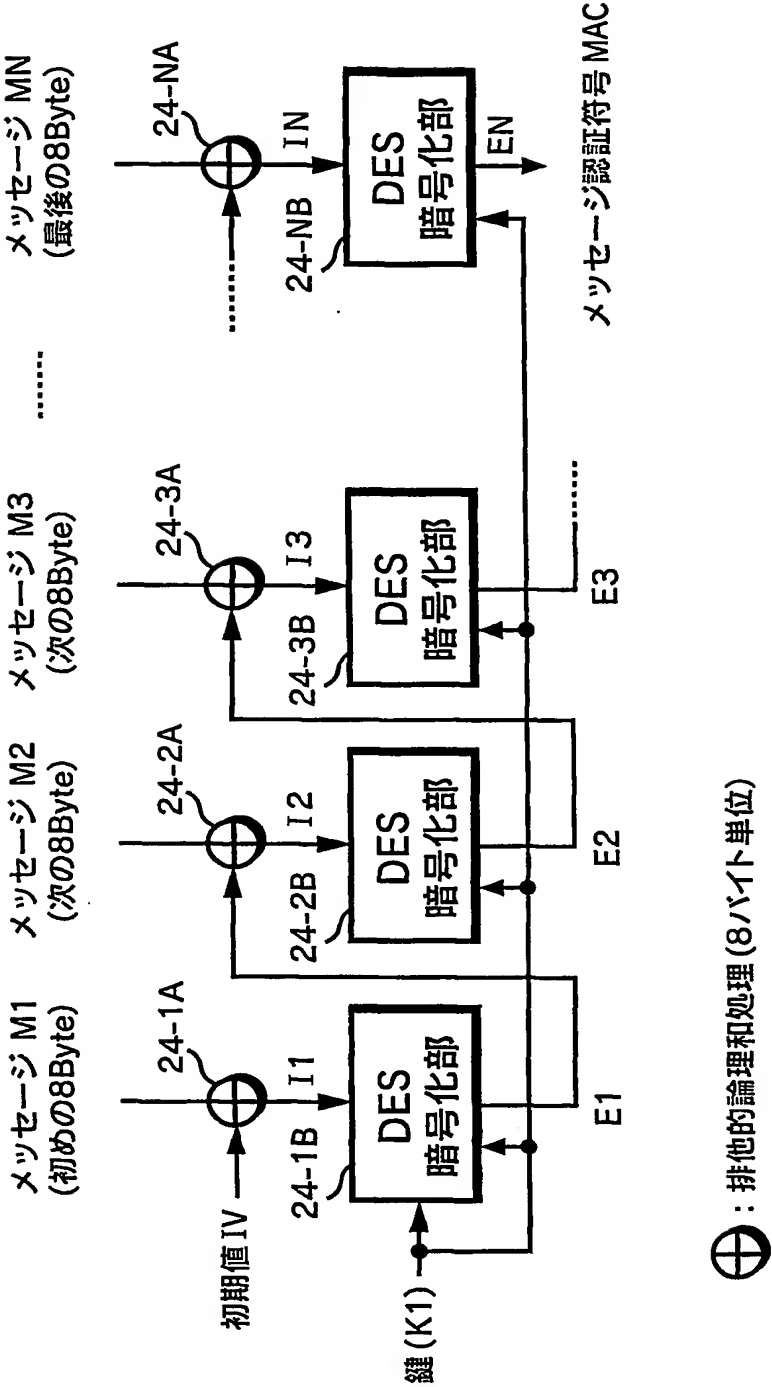
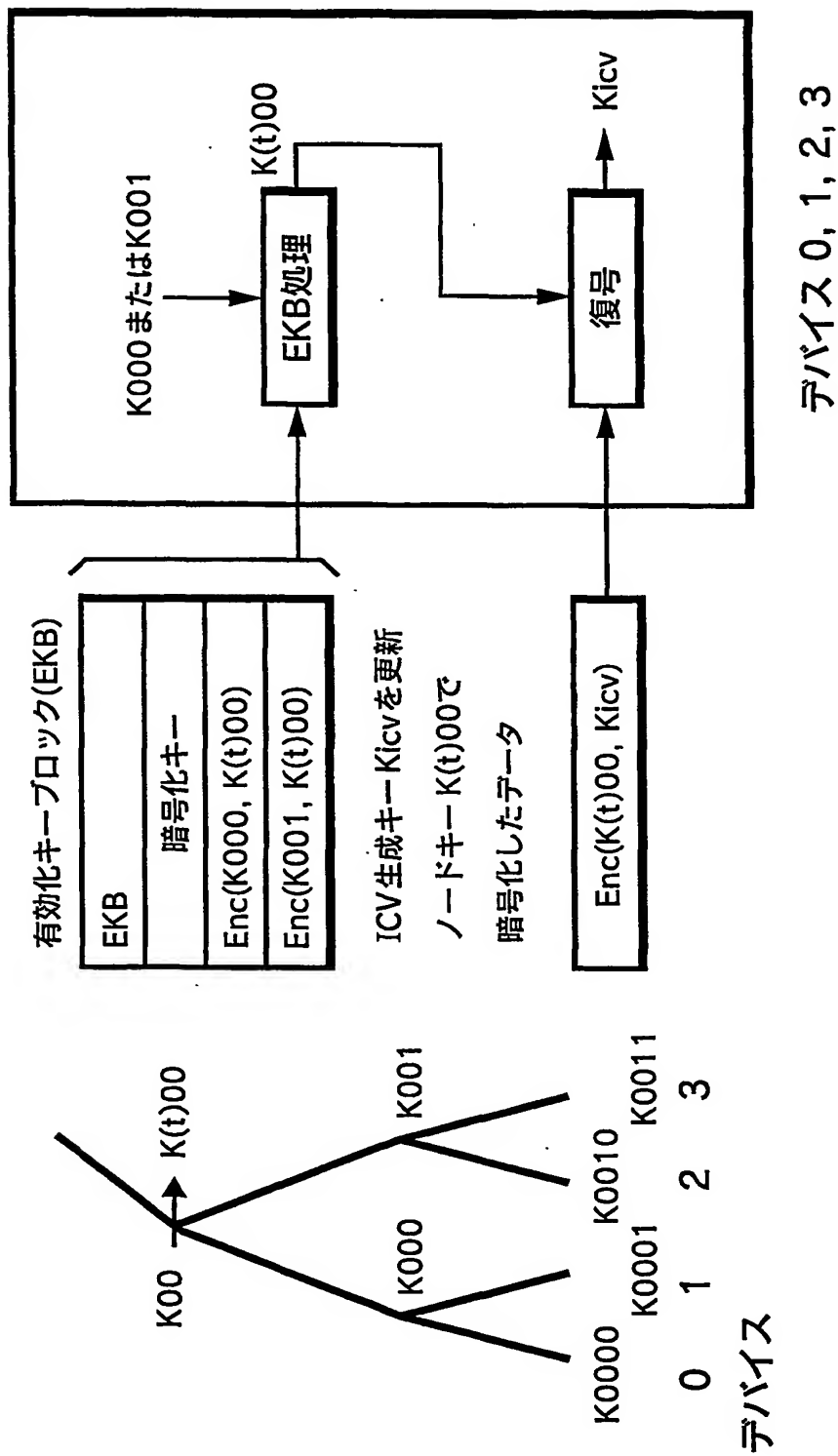
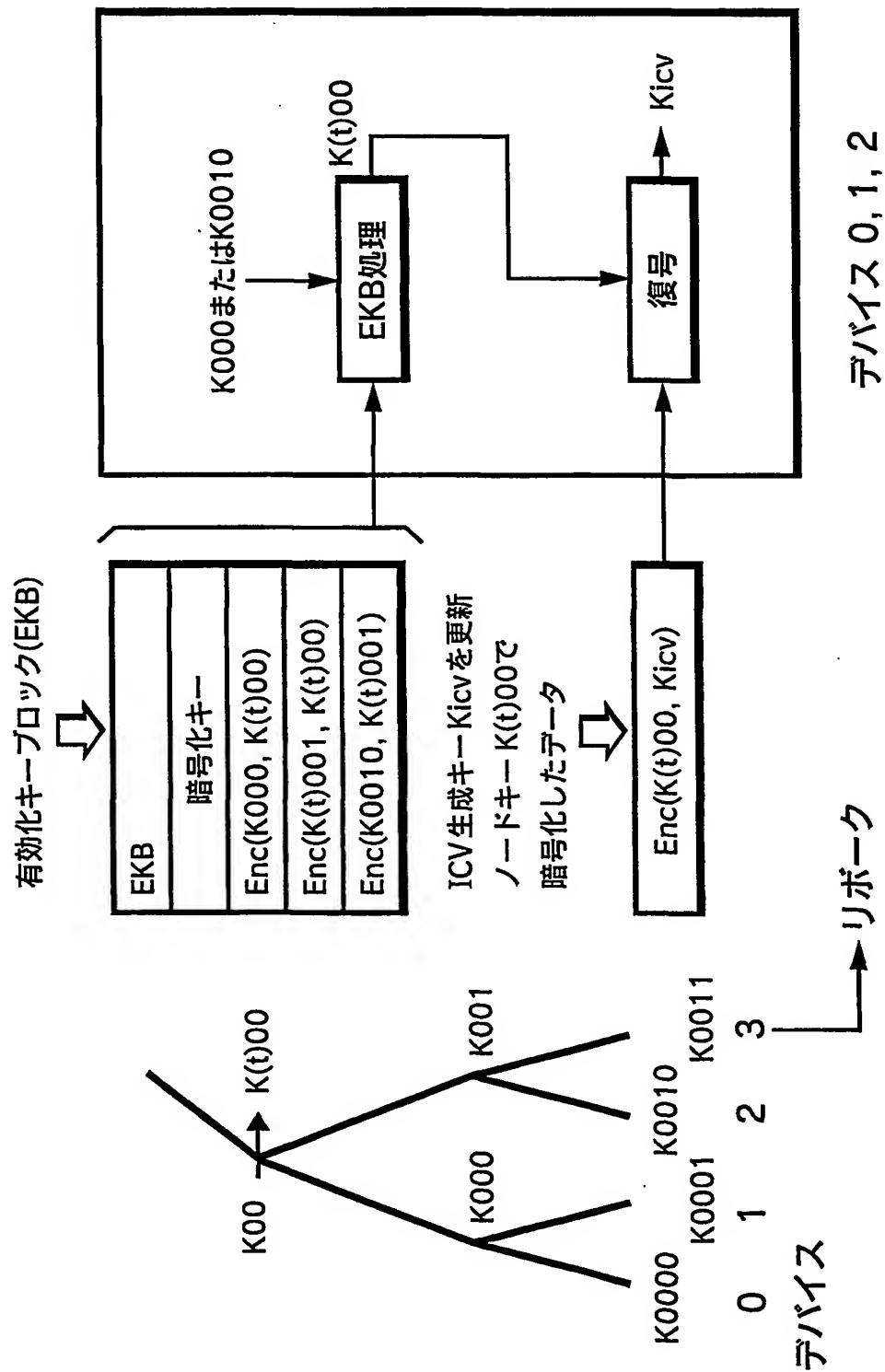


図 48

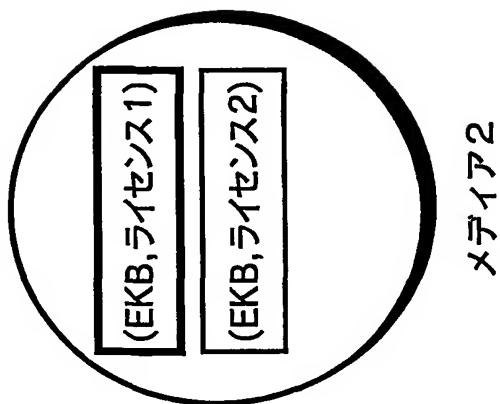


43/45

図 49

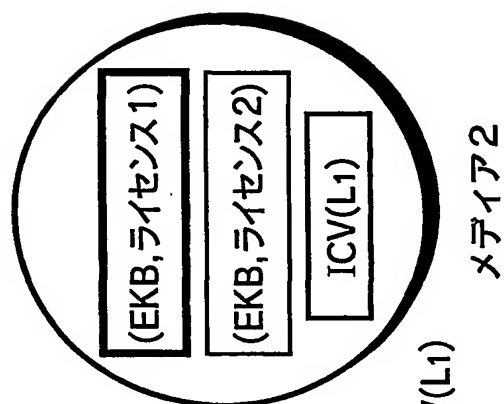


44/45



コピー

図50A

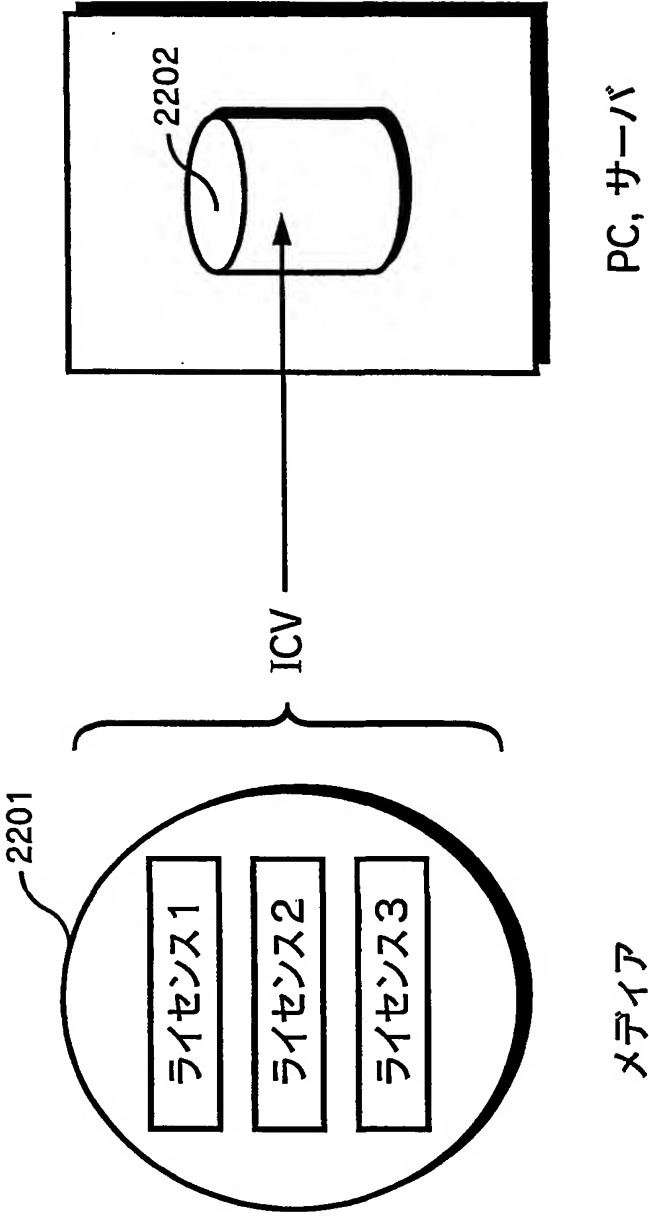


コピー

ICV(L1, L2) ≠ ICV(L1)

図50B

図51



PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT (PCT Article 17(2)(a), Rules 13ter.1(c) and 39)

Applicant's or agent's file reference 872-S02P0379	IMPORTANT DECLARATION	Date of mailing (<i>day/month/year</i>) 09 July, 2002 (09.07.02)
International application No. PCT/JP02/02959	International filing date (<i>day/month/year</i>) 27 March, 2002 (27.03.02)	(Earliest) Priority Date (<i>day/month/year</i>) 29 March, 2001 (29.03.01)
International Patent Classification (IPC) or both national classification and IPC Int.Cl ⁷ G06F17/60		
Applicant Sony Corp.		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below.

1. ☐ The subject matter of the international application relates to:
 - a. ☐ scientific theories.
 - b. ☐ mathematical theories.
 - c. ☐ plant varieties.
 - d. ☐ animal varieties.
 - e. ☐ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes.
 - f. ☐ schemes, rules or methods of doing business.
 - g. ☐ schemes, rules or methods of performing purely mental acts.
 - h. ☐ schemes, rules or methods of playing games.
 - i. ☐ methods for treatment of the human body by surgery or therapy.
 - j. ☐ methods for treatment of the animal body by surgery or therapy.
 - k. ☐ diagnostic methods practised on the human or animal body.
 - l. ☐ mere presentations of information.
 - m. ☐ computer programs for which this International Searching Authority is not equipped to search prior art.
2. ☒ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:

☐ the description
 ☒ the claims
 ☐ the drawings
3. ☐ The failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions prevents a meaningful search from being carried out :

☐ the written form has not been furnished or does not comply with the standard.
 ☐ the computer readable form has not been furnished or does not comply with the standard.
4. Further comments:
 The constituent elements stated in claims 1-5 are not supported at all by the description, and the inventions of claims 1-5 cannot be understood.

Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

特 許 協 力 条 約

P C T

国際調査報告を作成しない旨の決定

(法第8条第2項、法施行規則第42条、第50条の3第
〔P C T 17条(2) (a)、P C T 規則13の3.1(c)、39〕

出願人又は代理人 の書類記号 872-S02P0379	重要決定	発送日 (日.月.年) 09.07.02
国際出願番号 P C T / J P 0 2 / 0 2 9 5 9	国際出願日 (日.月.年) 27.03.02	優先日 (日.月.年) 29.03.01
国際特許分類 (I P C) I n t . C l ⁷ G 0 6 F 1 7 / 6 0		
出願人 (氏名又は名称) ソニー株式会社		

この出願については、法第8条第2項 (P C T 17条(2) (a)) の規定に基づき、次の理由により国際調査報告を作成しない旨の決定をする。

1. ☐ この国際出願は、次の事項を内容としている。
 - a. ☐ 科学の理論
 - b. ☐ 数学の理論
 - c. ☐ 植物の品種
 - d. ☐ 動物の品種
 - e. ☐ 植物及び動物の生産の本質的に生物学的な方法 (微生物学的方法による生産物及び微生物学的方法を除く。)
 - f. ☐ 事業活動に関する計画、法則又は方法
 - g. ☐ 純粋に精神的な行為の遂行に関する計画、法則又は方法
 - h. ☐ 遊戯に関する計画、法則又は方法
 - i. ☐ 人の身体の手術又は治療による処置方法
 - j. ☐ 動物の身体の手術又は治療による処置方法
 - k. ☐ 人又は動物の身体の診断方法
 - l. ☐ 情報の単なる提示
 - m. ☐ この国際調査機関が先行技術を調査できないコンピューター・プログラム
2. ☒ この国際出願の次の部分が所定の要件を満たしていないので、有効な国際調査をすることができない。

☐ 明細書
 ☒ 請求の範囲
 ☐ 図面
3. ☐ヌクレオチド又はアミノ酸の配列表が実施細則の附属書C (塩基配列又はアミノ酸配列を含む明細書等の作成のためのガイドライン) に定める基準を満たしていないので、有効な国際調査をすることができない。

☐ 書面による配列表が提出されていない又は所定の基準を満たしていない。
 ☐ 磁気ディスクによる配列表が提出されていない又は所定の基準を満たしていない。
4. 附記
請求の範囲 1 - 5 における各構成要素が、明細書のいずれの記載に対応するのか全く不明であり、請求の範囲 1 - 5 に係る発明を把握することができない。

名称及びあて名 日本国特許庁 (I S A / J P) 郵便番号 100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官 (権限のある職員) 谷口 信行 電話番号 03-3581-1101 内線 3560	5 L 9 4 6 7
--	--	-------------